# Advanced RFID-Based Footstep Power Harvesting and Charging System

A report submitted to the

Department of Electrical and Information Engineering
Faculty of Engineering
University of Ruhuna
Sri Lanka

On 09th of October 2024

In completing the group project for the module
EE6304 Embedded Systems Design

Group Members:

**EG/2020/3917**

**EG/2020/3919**

**EG/2020/3929**

# Table of Contents

## List of Figures

# 1 Introduction

## 1.1 Introduction to the Problem

In today's world most of human activities involves electricity. Generating enough power to cater these requirements have been a challenge for humans forever. What if we can use some form of wasted energy to fulfil this electricity requirement of certain applications? This project is an exact solution like that.

Here the energy usually wasted through our footsteps is utilized to charge an electrical appliance, specifically a mobile phone. Among the electrical appliances used by a person the mobile phone is probably the most used in a day. So, a typical phone must be charged at least two times per day for best performance, but we don't always get the requirements to do this. For example, a power outlet might not be in the surrounding even if it did it might not match the charger's pins. Therefore, instead of using a charger we can charge our mobile phones directly by our feet using this system.

As this uses the force given by the foot to charge it don't need a power outlet. For this project RFID technology is used for user identification and piezo sensors are used to generate the required electricity by trampling them and this electricity will be stored in a battery or can be used directly to charge the mobile phone. This can be used to power up other rechargeable appliances too. This device can be implemented in public places like bus stands, parks, malls etc. Specially for highly populated countries this system would be very useful.

## 1.2 Existing Solutions and their Drawbacks

Since this project focus on generating power to charge a small device like a mobile phone available solutions for this problem and some similar existing solutions solution will be discussed in this section,

1. **Traditional Charging Docks** – Traditionally people charge mobile phones using charging docks. To use this method the user, need the access to a power outlet which might not be available all the time. Also, any user with similar port can use these chargers hence they have the risk of being stolen due to lack of authentication.
2. **Pavegen Systems** – This is another example of energy harvesting using foot traffic. It utilizes a flooring system embedded with kinetic tiles that generate electricity when people walk on them. But these systems have issues such as high cost, low energy output and regular maintenance due to frequent wear and tear because of large foot traffic.
3. **Solar-Powered** - Solar chargers are used for powering small appliances in off-grid environments. But these devices are suitable only for low power applications because low power output. Also, on a gloomy day its difficult to get sufficient power generation.

4. **Human Power Kinetics –** This is very similar to footstep charging but unlike footstep energy harvesting, which utilizes natural movement, hand-crank generators require the user to exert focused effort, making it impractical for continuous charging.

## 1.3 Objectives

The objective of this project are as follows,

1. **Energy Harvesting Efficiency**

   The main objective of this project is to develop an efficient system to harvest energy generated from foot pressure to generate electricity which can be used to charge or powerup an electrical appliance.
2. **User Identification and Authentication**

   With the aid of RFID technology, the system is given authorized access to protect the device from being misused or stolen by unauthorized users.
3. **Power Storage and Management**

   Further in this project we have designed and effective energy storage solution to store the harvested energy so the user can use this energy for later use.
4. **Portability and Convenience**

   As this device is small and easy to use the user can carry it anywhere easily. Upon further development this device can be integrated to a show so or made even smaller increasing its portability.
5. **Eco-friendliness**

   As this is a completely environmentally friendly and renewable energy source our project promotes environmental sustainability.
6. **Deployment in Public Spaces.**

   This mechanism can also be used in public areas. For example, by installing this in malls, schools or streets where many people often walk, the generated energy from the footstep pressure of the people can be used to light up streetlamps, energize a decorating device or powerup some other devices as well.

7. **User interface and Feedback**

   In this project an LCD display is integrated to give real-time feedback to the users. So, if the user is authorized it will indicate ‘Authorized user’ if not it’ll indicate ‘Unauthorized user’. Also, the LCD is used to tell the user about the charging status,

## 1.4 Scope of the Project

The scope of this project is to design and develop a **Footstep Energy Harvesting System** that harnesses kinetic energy from human footsteps using piezoelectric sensors to charge mobile phones and other rechargeable devices. The system will incorporate **RFID technology for user authentication.** [1]

For this project we are making a prototype that can generate electricity using piezoelectric sensors. This generated electricity can be stored in a battery and be used to charge a mobile phone. Furthermore, the prototype has an RFID authentication system to provide authorized access. An LCD display is used to give a user interface for user interacting.

However the project has limitations due to the efficiency of the piezoelectric sensors, which sometimes generate small amounts of energy which is not enough to powerup high-energy appliances. Therefore, this device is limited for usage of low power operations such as charging a mobile phone. Also, the system's reliability depends on the quality of the RFID reader and piezoelectric materials so low-quality or damaged components will cause delays and failures. Furthermore, the budget limitations have restricted us from using advanced RFID readers and better energy storage elements such as superconductors. The project does not include any advanced security features beyond basic RFID authentication which might me a potential threat for user's privacy.

For usage of this device in public areas we have assumed that enough foot traffic will be there to implement the device. Also, we have assumed that piezoelectric sensors will be durable enough to withstand constant pressure over long periods and the RFID system will reliably authenticate users without significant interference from other electronic devices or environmental factors.

# 2 Specifications

**1. RFID-Based User Authentication**

This is to ensure only authorized users can use the system to charge their device. RFID module is used to identify the users. This ensures the security of the system. Successful authentication will be indicated by a green LED, while failed authentication will trigger a red LED.

**2. Piezoelectric Energy Harvesting**

An array of piezoelectric sensors is used to harvest the energy from footsteps (any other pressure applied) effectively [2]. This harvested energy should be sufficient to charge the mobile phone or another small rechargeable device. Our goal is to get 5V from this by amplification.

**3. Efficient Voltage Regulation**

Voltage regulator is used to regulate the voltage to required forms. This is done to ensure the safety for the mobile phone or device connected.

**4. Real-Time User Feedback**

An LCD display provides real-time feedback on the charging status. It will show a message indicating 'Charging' and also some messages.

**5. Power Storage Capability**

A battery or another power storing element can be integrated to store power to be used later. A device like a power bank would be good for this. For this we are using a rechargeable battery.

**6. Portability and Ease of Installation**

The final product should be portable, so the user can carry it anywhere easily. Also, it should be easy to install in public areas for general use. Due to resource availability and ease of demonstration our prototype is made in a bigger size, but this can be made more portable by further development.

**7. Environmental and Energy Efficiency**

The system is designed to maximize energy efficiency by using wasted energy from footsteps to good use. As it uses renewable energy principles, it contributes to reducing the overall carbon footprint.

**8. User-Friendly Operation**

With very less requirement this system serves as a very user-friendly operation. As this is a very simple system any user can use this approach to charge their phones with minimum technical knowledge. Furthermore, the usage of LCD and LEDs makes the user experience even more user-friendly.

# 3 Implemented Solution

## 3.1 Block Diagram

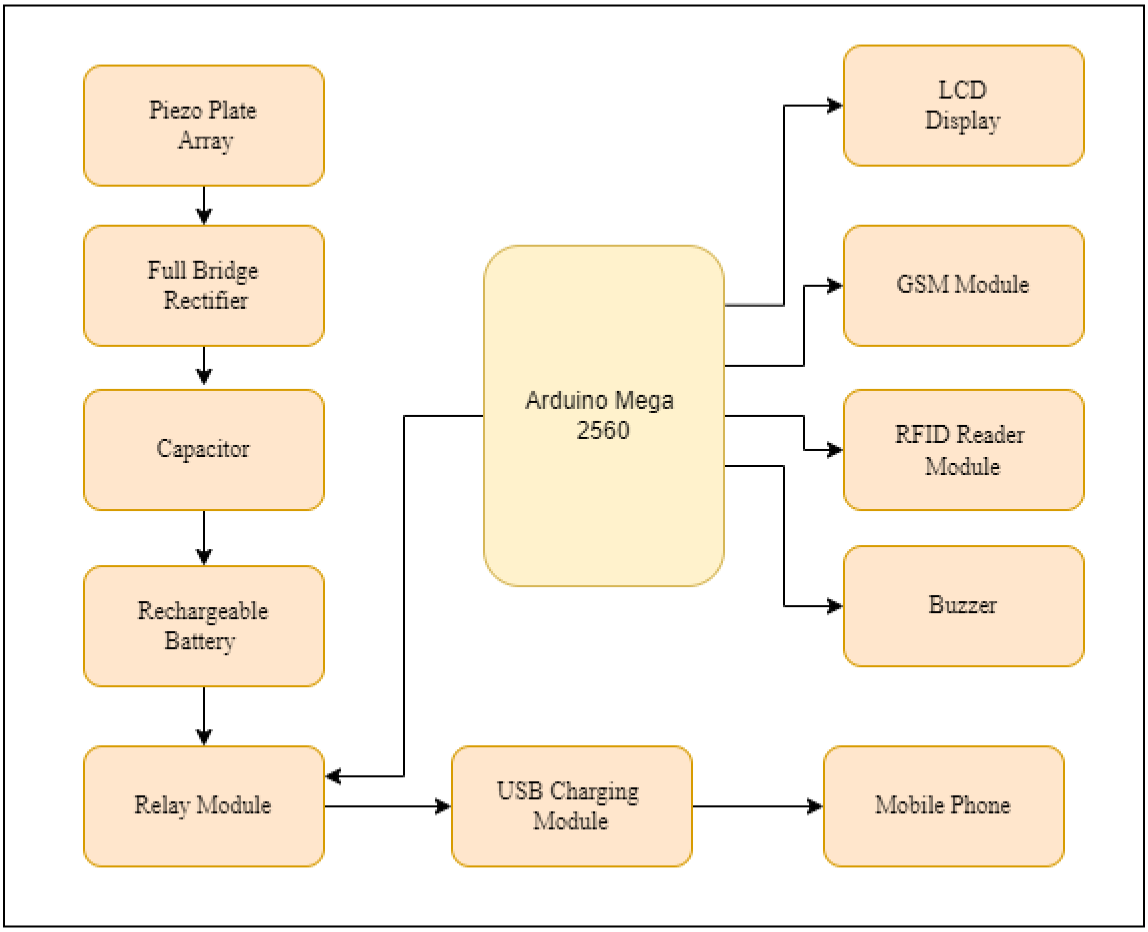


Figure 1: Block diagram of the final project

## 3.2 Block Specifications

### 1. Microcontroller (Atmega2560)

ATmega2560 microcontroller is used as the central processing unit controlling and coordinating all components. It is used to control many of the blocks and peripherals in the system. The functions it controls includes obtaining authentication from RFID block, using authentication data to control the relay, displaying content in the LCD display and also the controlling of the GSM module.

### 2. Piezo Plate Array

An array of parallelly and series connect piezo plates is used to generate electricity. A piezoelectric sensor has the ability to generate electricity upon a mechanical stress given to it. A single sensor can generate voltage ranging from 1 to 15 V.

### 3. Full Bridge Rectifier

Thus, the amplified AC voltage is sent through a full bridge rectifier to be converted to DC

### 4. Relay

Relay is acting as a switch here which allows and prohibits the charging process based on the signal given by the RFID authentication.

### 5. Rechargeable Battery

A rechargeable battery is used to store the voltage to provide the voltage to the mobile phone in consistent manner.

### 6. USB Charging Module

USB charging module is used to connect the circuit to the mobile phone or another rechargeable device.

### 7. LCD Display

LCD Display is used to display the charging status of the mobile phone. While its charging LCD will display ‘Charging’. Once fully charged it’ll display ‘Fully Charged’.

### 8. RFID Reader and Cards (RDM6300 RFID/ EM18 RFID)

For this project the RDM6300 is used as the reader. It’s programmed to only authorize certain set of card IDs. Therefore, it’ll signal the relay to switch on only when an authorized user is detected.

### 9. GSM Module

A GSM module is used to send a message to the user once charging is completed. It can be programmed to send any other message too.

### 10. Buzzer

The buzzer is used to alert the if an unauthorized user is trying to access the system

## 3.3 Model solution

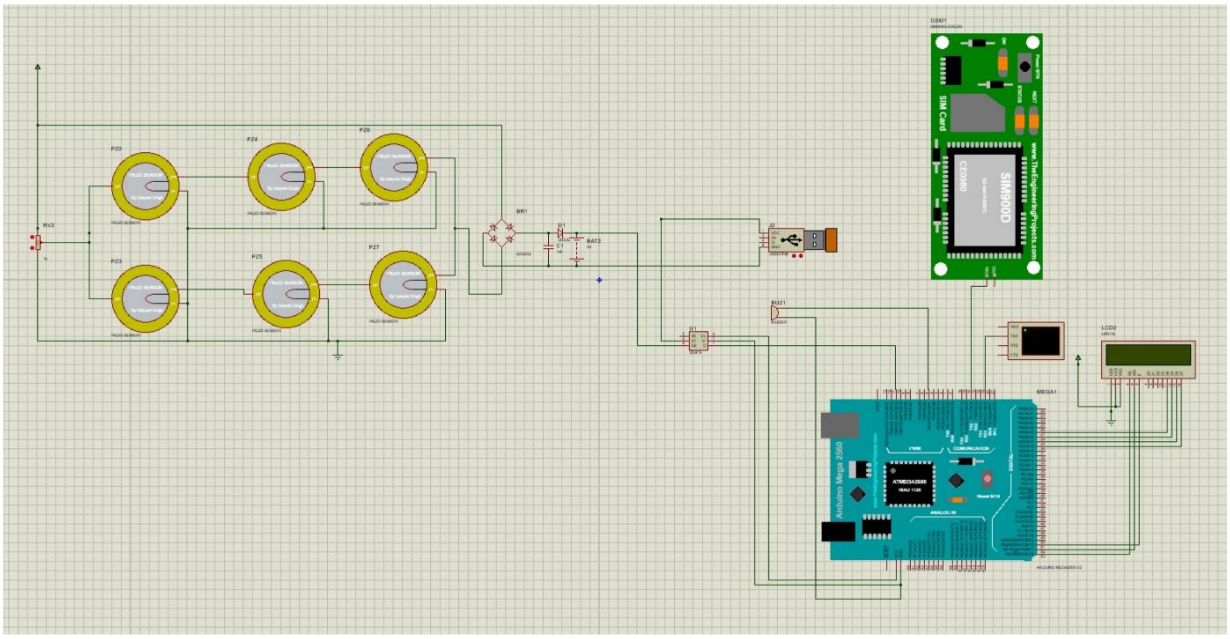

Figure 2: Proteus Simulation Diagram

The Figure 2 diagram illustrates the Proteus simulation of the circuit. In the simulation, a virtual terminal is used in place of the RDM6300 RFID reader, which is connected to the RX 1 port. While in Proteus the input is manually typed, in a real-world scenario, the RFID tag can simply be scanned by holding the card in front of the reader module.

### 3.3.1 Code

**main.c**

```
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include "lcd.h"

#define BUFFER_SIZE 14
#define DATA_SIZE 10
#define DATA_TAG_SIZE 8
#define READ_LED_PIN PE5
#define CHARGE_PIN PH5
#define RELAY_PIN PE3
#define ACCESS_GRANTED_TAG_ID 0x39123B

volatile uint8_t buffer[BUFFER_SIZE];
volatile int buffer_index = 0;
```

```c
void UART_Init(unsigned int ubrr);
unsigned char UART_Receive(void);
unsigned long extract_tag(void);
unsigned long hexstr_to_value(char *str, unsigned int length);

int main(void) {
	// Initialize LCD
	initLcdPorts();
	Lcd_init();

	// Set READ_LED_PIN, RELAY_PIN, and CHARGE_PIN as output
	DDRE |= (1 << READ_LED_PIN) | (1 << RELAY_PIN);
	DDRH |= (1 << CHARGE_PIN);

	// Turn off relay initially (ensure low output)
	PORTE |= (1 << RELAY_PIN);

	// Initialize UART with baud rate 9600
	UART_Init(103); // 9600 baud for 16 MHz clock

	// Display welcome message
	Lcd_CmdWrite(0x80);
	Lcd_displayMessage("Welcome to ESD");
	_delay_ms(1000);
	Lcd_CmdWrite(0xC0);
	Lcd_displayMessage("Group 7");
	_delay_ms(2000);
	Lcd_clear();

	// Display instruction to place access card
	Lcd_CmdWrite(0x80);
	Lcd_displayMessage("Place your access");
	Lcd_CmdWrite(0xC0);
	Lcd_displayMessage("card");
	_delay_ms(1000);

	while (1) {
		if (UCSR0A & (1 << RXC0)) {
			bool call_extract_tag = false;
			int ssvalue = UART_Receive();

			if (ssvalue == 2) {
				buffer_index = 0;
				} else if (ssvalue == 3) {
				call_extract_tag = true;
			}

			if (buffer_index >= BUFFER_SIZE) {
				buffer_index = 0;
				continue;
			}

			buffer[buffer_index++] = ssvalue;

			if (call_extract_tag && buffer_index == BUFFER_SIZE) {
				unsigned long tag_id = extract_tag();

				Lcd_clear();
				Lcd_CmdWrite(0x80);
				if (tag_id == ACCESS_GRANTED_TAG_ID) {
					Lcd_displayMessage("Access Granted");
					_delay_ms(1000);
					Lcd_clear();
					Lcd_CmdWrite(0x80);
					Lcd_displayMessage("Welcome Vidyan");
					Lcd_CmdWrite(0xC0);
					Lcd_displayMessage("Initiating...");
```

```
                                        // Activate relay for charging
                                        PORTE &= ~(1 << RELAY_PIN);

                                        _delay_ms(1000);
                                        Lcd_clear();
                                        Lcd_CmdWrite(0x80);
                                        Lcd_displayMessage("Charging...");
                                        Lcd_CmdWrite(0xC0);
                                        Lcd_displayMessage("You have 50s");
                                        _delay_ms(50000);  // Simulate 50 seconds of charging

                                        // Turn off relay after charging is done
                                        PORTE |= (1 << RELAY_PIN);

                                        Lcd_clear();
                                        Lcd_CmdWrite(0x80);
                                        Lcd_displayMessage("Disconnecting...");
                                        _delay_ms(2000);

                                        // Turn on READ_LED_PIN to signal done
                                        PORTE &= ~(1 << READ_LED_PIN);
                                        } else {
                                        Lcd_displayMessage("Access Denied");
                                        PORTE |= (1 << READ_LED_PIN);
                                }

                                _delay_ms(2000);

                                // Reset display to show instruction
                                Lcd_clear();
                                Lcd_CmdWrite(0x80);
                                Lcd_displayMessage("Place your access");
                                Lcd_CmdWrite(0xC0);
                                Lcd_displayMessage("card");

                                buffer_index = 0;
                        }
                }
        }
}

void UART_Init(unsigned int ubrr) {
        UBRR0H = (unsigned char)(ubrr >> 8);
        UBRR0L = (unsigned char)ubrr;
        UCSR0B = (1 << RXEN0) | (1 << TXEN0);
        UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);
}

unsigned char UART_Receive(void) {
        while (!(UCSR0A & (1 << RXC0)));
        return UDR0;
}

unsigned long extract_tag(void) {
        uint8_t *msg_data_tag = buffer + 3;
        return hexstr_to_value((char *)msg_data_tag, DATA_TAG_SIZE);
}

unsigned long hexstr_to_value(char *str, unsigned int length) {
        char copy[length + 1];
        for (int i = 0; i < length; i++) {
                copy[i] = str[i];
        }
        copy[length] = '\0';
        return strtoul(copy, NULL, 16);
}
```

**lcd.h**

```c
#ifndef LCD_H_
#define LCD_H_


#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

#define LcdDataBus PORTA
#define LcdControlBus PORTA
#define LcdDataBusDirnReg DDRA

#define LCD_RS 0
#define LCD_RW 1
#define LCD_EN 2

void initLcdPorts(void)
{
	LcdDataBusDirnReg |= 0xF0; // LCD Data pins to o/p (PD4-PD7)
	DDRA |= 0x07;              // LCD control pins (RS, RW, EN) as output
}

void Lcd_CmdWrite(char cmd)
{
	LcdDataBus = (cmd & 0xF0);           // Set upper 4 bits of the command
	LcdControlBus &= ~(1 << LCD_RS);      // Set RS pin to LOW
	LcdControlBus &= ~(1 << LCD_RW);      // Set RW pin to LOW
	LcdControlBus |= (1 << LCD_EN);       // Generate High-to-Low pulse
	_delay_ms(10);
	LcdControlBus &= ~(1 << LCD_EN);      // Clear EN pin

	_delay_ms(10);

	LcdDataBus = ((cmd << 4) & 0xF0);     // Set lower 4 bits of the command
	LcdControlBus &= ~(1 << LCD_RS);      // Set RS pin to LOW
	LcdControlBus &= ~(1 << LCD_RW);      // Set RW pin to LOW
	LcdControlBus |= (1 << LCD_EN);       // Generate High-to-Low pulse
	_delay_ms(10);
	LcdControlBus &= ~(1 << LCD_EN);      // Clear EN pin

	_delay_ms(10);
}

void Lcd_DataWrite(char dat)
{
	LcdDataBus = (dat & 0xF0);            // Set upper 4 bits of the data
	LcdControlBus |= (1 << LCD_RS);       // Set RS pin to HIGH
	LcdControlBus &= ~(1 << LCD_RW);      // Set RW pin to LOW
	LcdControlBus |= (1 << LCD_EN);       // Generate High-to-Low pulse
	_delay_ms(10);
	LcdControlBus &= ~(1 << LCD_EN);      // Clear EN pin

	_delay_ms(10);

	LcdDataBus = ((dat << 4) & 0xF0);     // Set lower 4 bits of the data
	LcdControlBus |= (1 << LCD_RS);       // Set RS pin to HIGH
	LcdControlBus &= ~(1 << LCD_RW);      // Set RW pin to LOW
	LcdControlBus |= (1 << LCD_EN);       // Generate High-to-Low pulse
	_delay_ms(10);
	LcdControlBus &= ~(1 << LCD_EN);      // Clear EN pin

	_delay_ms(10);
}
```

```c
void Lcd_init(void)
{
	LcdControlBus &= ~(1 << LCD_RW);  // Ensure RW is LOW (write mode)
	_delay_ms(100);
	Lcd_CmdWrite(0x02);       // Initialize in 4-bit mode
	_delay_ms(5);
	Lcd_CmdWrite(0x28);       // 4-bit, 2-line, 5x7 format
	_delay_ms(5);
	Lcd_CmdWrite(0x0C);       // Display ON, Cursor OFF
	_delay_ms(5);
	Lcd_CmdWrite(0x06);       // Increment cursor to right
	_delay_ms(5);
	Lcd_CmdWrite(0x01);       // Clear display
	_delay_ms(5);
}

void Lcd_clear(void)
{
	Lcd_CmdWrite(0x01);       // Clear the LCD display
	_delay_ms(10);
}

void Lcd_displayMessage(const char* message)
{
	while (*message)
	{
		Lcd_DataWrite(*message);  // Write each character to the LCD
		message++;
	}
}

void Lcd_shiftLeft(void)
{
	Lcd_CmdWrite(0x18);       // Shift display to the left
	_delay_ms(300);
}

void Lcd_shiftRight(void)
{
	Lcd_CmdWrite(0x1C);       // Shift display to the right
	_delay_ms(300);
}



#endif /* LCD_H_ */
```

**rfid.h**

```c
#ifndef RFID_H_
#define RFID_H_

#include "usart.h"

#ifndef bit_is_clear
#define bit_is_clear(sfr, bit) (!(sfr & (1 << bit)))
#endif

// RDM6300 frame constants
#define RFID_FRAME_SIZE 14
#define START_BYTE 0x02
#define END_BYTE 0x03
```

```c
// Function to read a valid 14-byte frame from RDM6300
int read_rdm6300_frame(char *rfid_tag) {
	char rfid_frame[RFID_FRAME_SIZE];
	uint8_t received_byte;
	int i = 0;

	// Wait for start byte (0x02)
	while ((received_byte = uart_receive()) != START_BYTE);

	// Read the next 12 bytes (10 UID ASCII + 2 checksum)
	for (i = 0; i < RFID_FRAME_SIZE - 2; i++) {
		rfid_frame[i] = uart_receive();
	}

	// Verify the end byte (0x03)
	if (uart_receive() != END_BYTE) {
		return 0; // Invalid frame
	}

	// Extract the 10 ASCII bytes representing the UID
	for (i = 0; i < 10; i++) {
		rfid_tag[i] = rfid_frame[i];
	}

	return 1; // Valid frame
}

#endif /* RFID_H_ */
```

## usart.h

```c
#ifndef USAR_H_
#define USAR_H_

#define F_CPU 16000000UL
#define BAUD 9600
#define BAUD_PRESCALE ((F_CPU / (BAUD * 16UL)) - 1)

void uart_init();
void uart_transmit(unsigned char data);
unsigned char uart_receive();

// Function to initialize UART communication
void uart_init() {
	// Set baud rate
	UBRR0H = (uint8_t)(BAUD_PRESCALE >> 8);
	UBRR0L = (uint8_t)(BAUD_PRESCALE);
	// Enable receiver and transmitter
	UCSR0B = (1 << RXEN0) | (1 << TXEN0);
	// Set frame format: 8 data bits, 1 stop bit, no parity
	UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);
}

// Function to receive a character via UART
unsigned char uart_receive() {
	// Wait for data to be received
	while (!(UCSR0A & (1 << RXC0)));
	// Get and return received data from buffer
	return UDR0;
}

#endif /* USAR_H_ *
```

### 3.3.2 Flow chart of the firmware

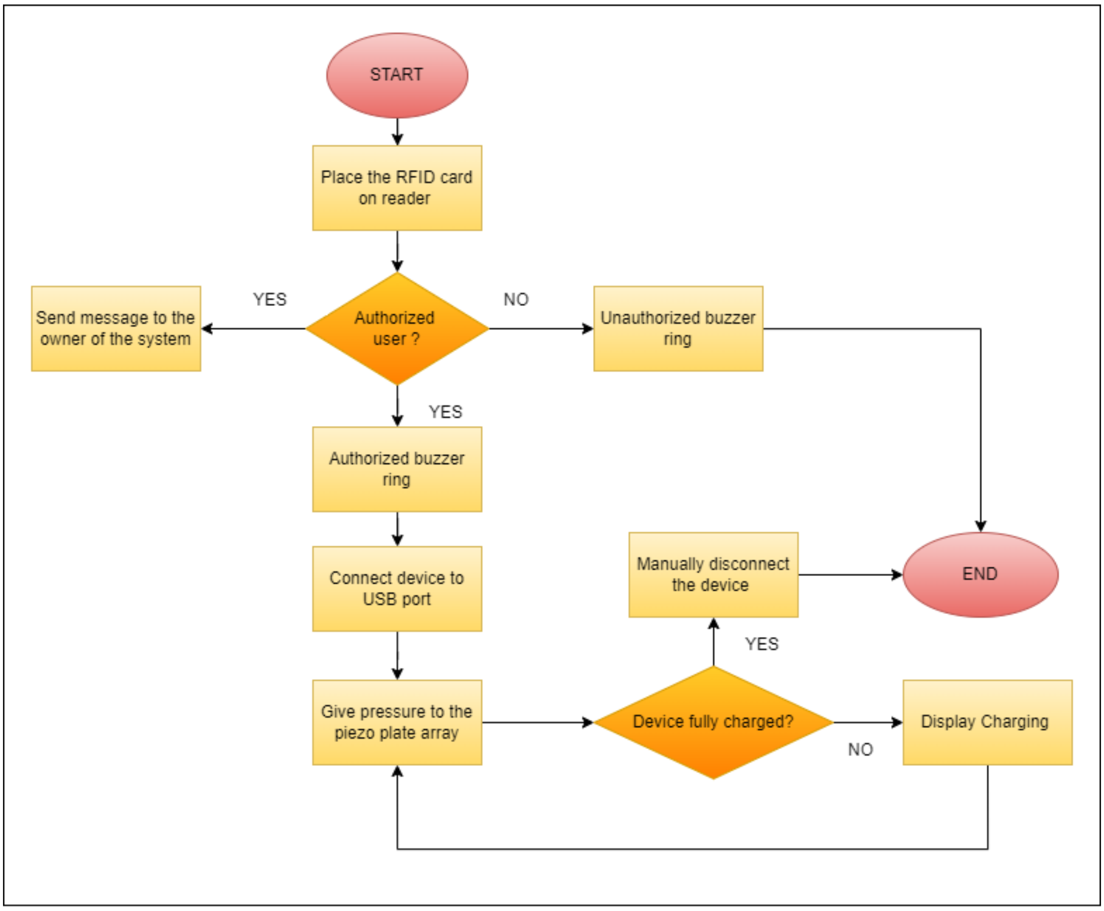


Figure 3: Flow Chart of the Firmware

The process in Figure 3 starts when a user places an RFID card on the reader. The system reads the RFID card to verify if the user is authorized or not. This is the first step in controlling access to the piezoelectric voltage generation system. If the user is not authorized, the system triggers an unauthorized buzzer sound, informing the user that access has been denied. At this point, no further actions are possible, and the process ends here.

If the user is authorized, the system triggers an authorized buzzer sound, allowing the user to proceed. In addition, a message is sent to the system's owner to notify them that an authorized user has accessed the system. This ensures that the system owner is kept informed about who is using the system.

Following successful authorization, the user can connect their device to the USB port. Afterward, they apply pressure to the piezoelectric plate array, which begins generating voltage through the piezoelectric effect. This voltage is used to charge the connected device. The system continues to monitor whether the device is fully charged. If the device is charged, the user is prompted to manually disconnect the device, and the process ends. If the device is not yet fully charged, a "charging" status is displayed, and the voltage generation continues until the device is fully charged.

In both cases of authorized and unauthorized access, the same buzzer is used, but it produces distinct sounds to differentiate between successful and failed access attempts.

# 4 Chapter 4: Implementation

## 4.1 Photographs during the process

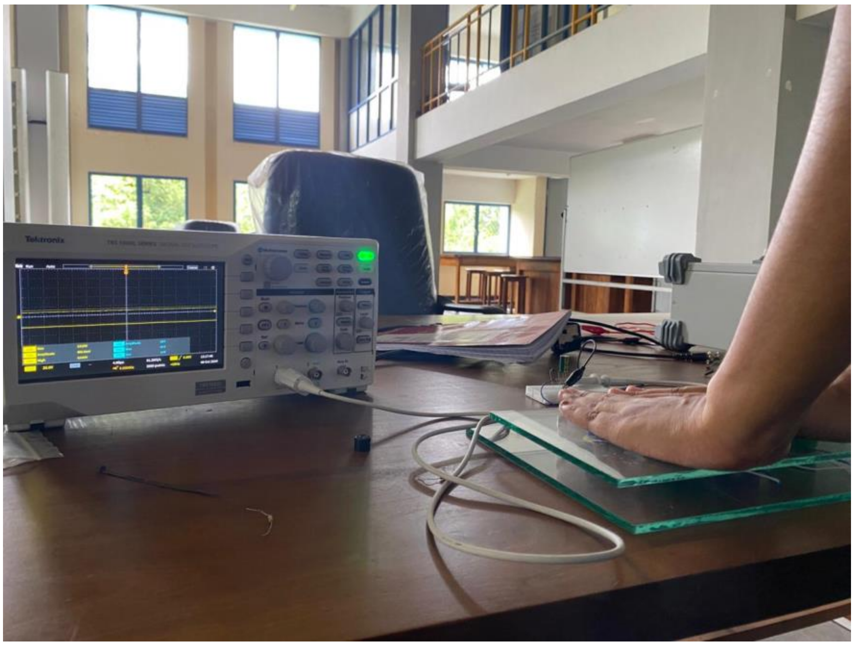

Figure 4: Testing power generation of the piezo plate array in the laboratory

Figure 5: Testing the charging of a rechargeable battery using an oscilloscope

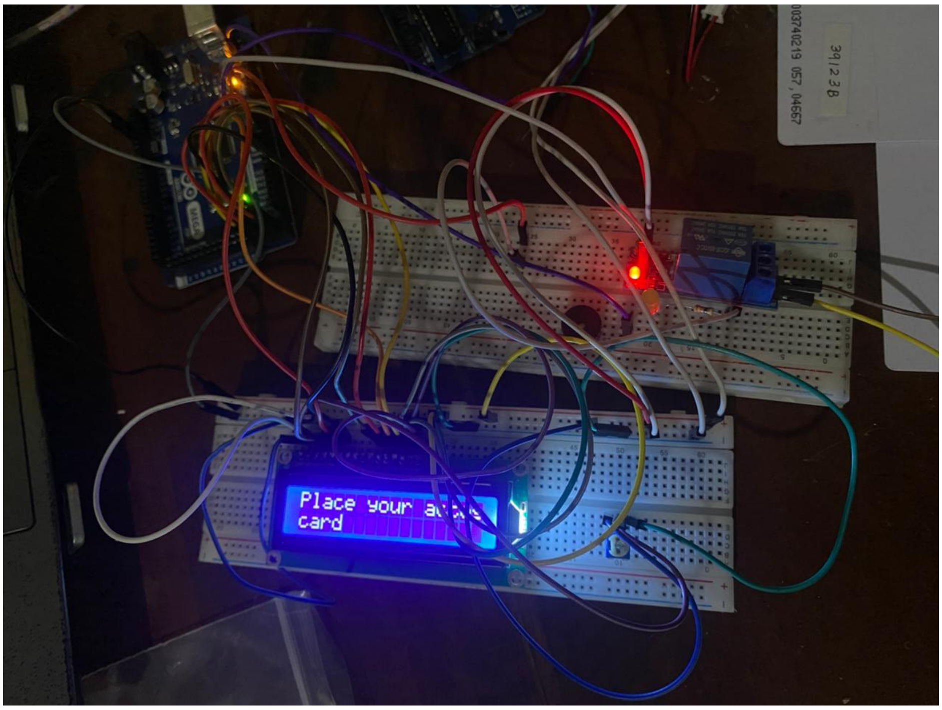


Figure 6: Implementation of an RFID module and LCD display for user verification before authentication using basic components.

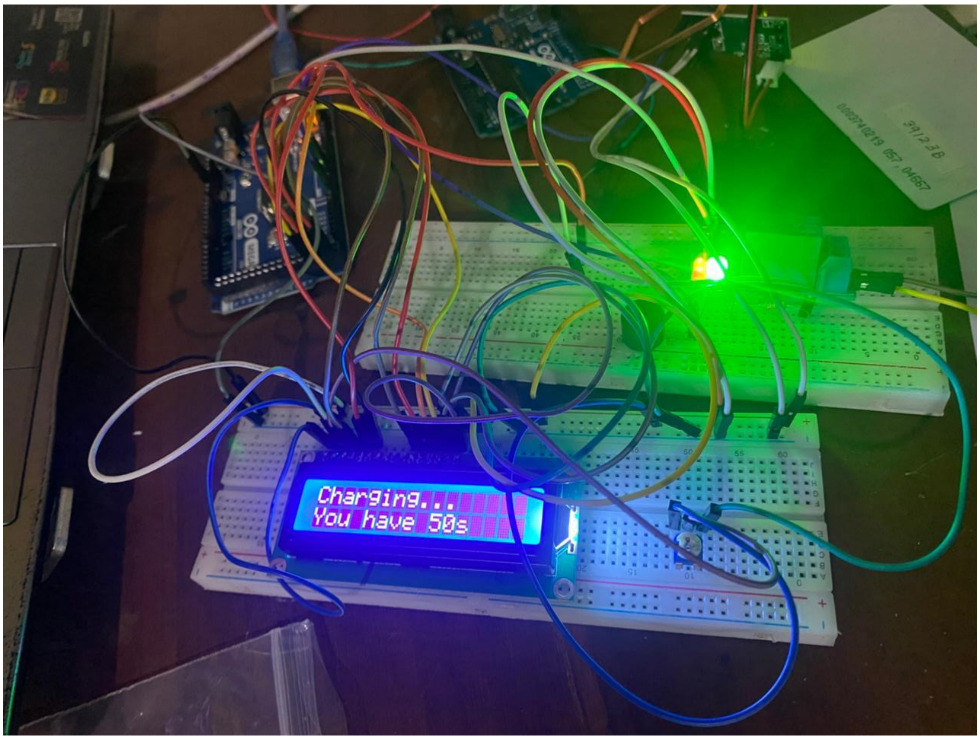


Figure 7: Testing the RFID module while a mobile phone is charging

# 5 References

[1] A. babu, "Autodesk Instructables," 26 January 2017. [Online]. Available: https://www.instructables.com/ADVANCED-FOOTSTEP-POWER-GENERATION-SYSTEM/. [Accessed 15 September 2024].

[2] S. PANGHATE, "Advanced Footstep Power Generation System using RFID for Charging," *International Research Journal of Engineering and Technology,* no. 07, p. 5, 2020.