# 🌀 FanTimer Project - Omar Abdelrahman Abbas

## 📜 Project Overview

**FanTimer** is an **Arduino-based project** designed to control a fan with a *user-defined timer*. Perfect for **automation** and everyday use! 💡

🛠️ **Created by:** Omar Abdelrahman Abbas on **12/10/2024**
🥈 **Department:** Software Engineering
🪪 **Code:** 162021603
👩‍🏫 **Supervisor:** Dr. Mostafa Mekky
💼 **Teaching Assistant:** Ahmed Abdelaziz

## 🔗 Relevant Files and Source Code:

- 📽️ **Preview Video:** youtu.be/WxNZjEOgeos
- 🛠️ **Adding H-Bridge Motor Driver:** youtu.be/jhzKlX_i4YQ
- 📂 **Google Drive:** drive.google.com/drive/folders/1t7pokOaPTi40edPa0rQoVFkKiX6PXEVr
- 📁 **GitHub Repository:** github.com/Omar7001-B/FanTimerArduio

## 🌟 Benefits and Use Cases

### 🌟 Benefits:

- 🏠 **Home Automation:** Easily schedule fan operation.
- 🌱 **Greenhouses:** Maintain optimal cooling for plants.
- ⚙️ **Workshops:** Improve air circulation and reduce fumes.

### 💡 Use Cases:

- 🌞 Cool a room before arrival.
- 🔥 Ventilate post-cooking odors.
- 🐾 Enhance pet comfort during hot days.
- 📚 Prepare spaces for events or study environments.

## 🎹 Keypad 4x4

A compact 4x4 keypad for user input, perfect for controlling timers and other applications. 🔢


## 📖 Usage Manual

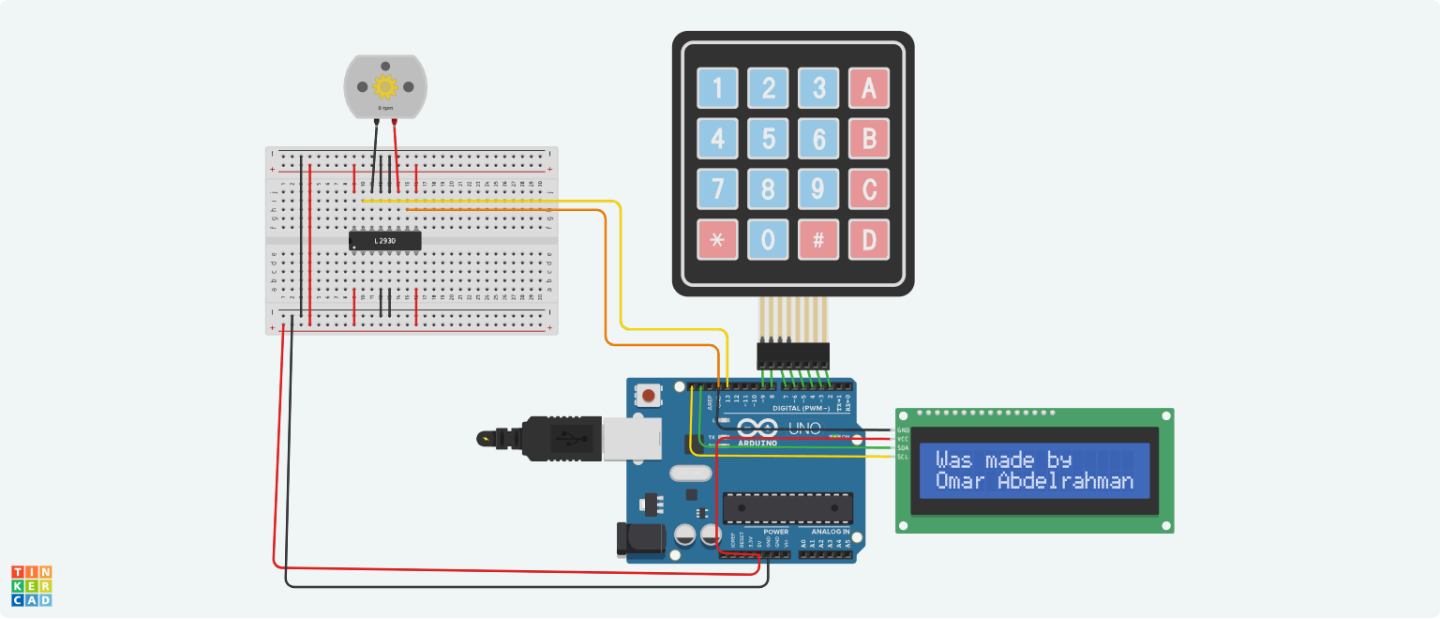
The **FanTimer** is controlled via a **4x4 Keypad**. Below are the control details:

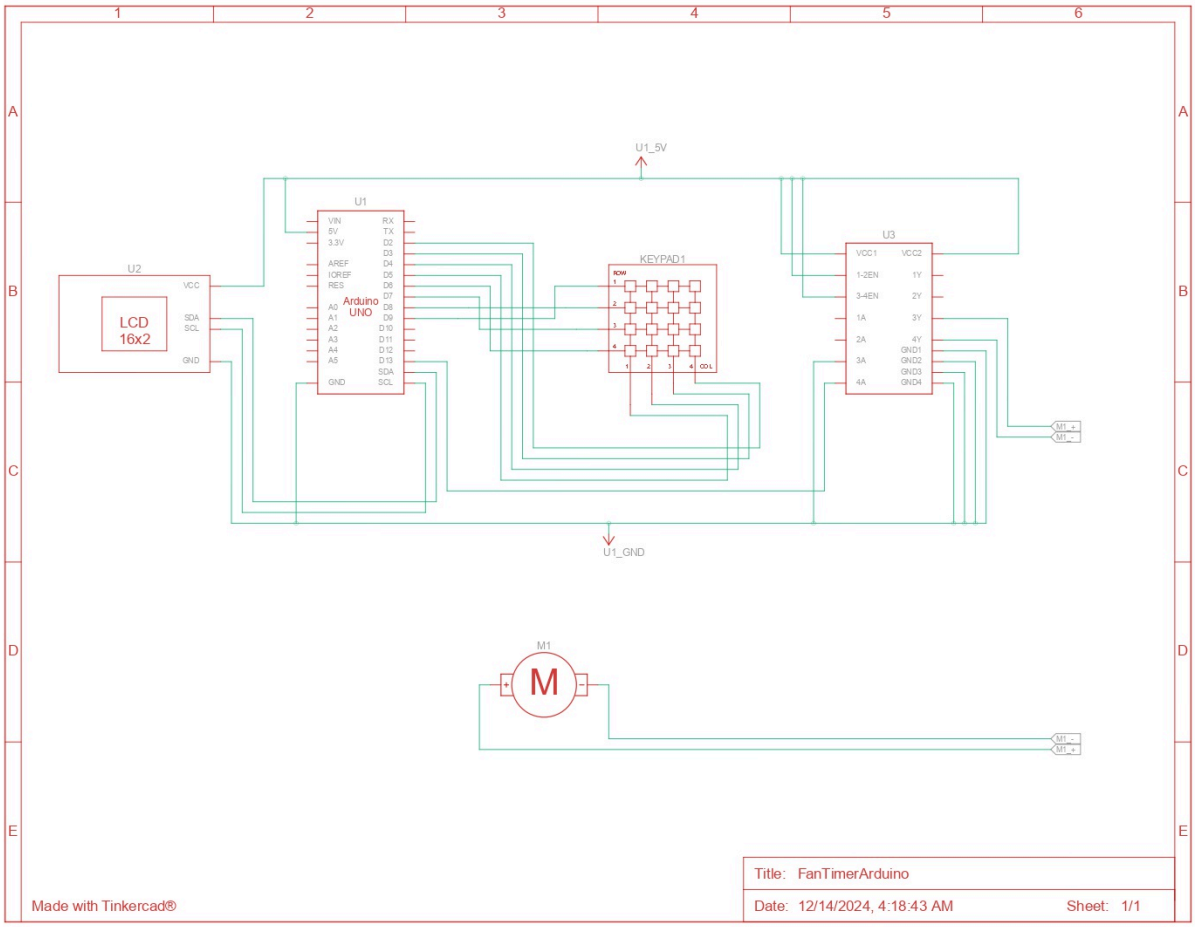
| Key | Function | Description |
|---|---|---|
| 0-9 | Enter Time | Set the timer duration (in seconds) |
| A | Start Timer | Begin the countdown |
| B | Pause/Resume | Toggle between pausing and resuming |
| C | Reset | Clear the current time and reset |
| D | Quick Add | Add 10 seconds to the timer |
| # | Show Credits | Display creator info |

## ⚙️ Basic Operation:

- ⏳ Use **numeric keys** to input the desired duration (seconds).
- ▶️ Press **'A'** to start, **'B'** to pause, and **'C'** to reset.
- ➕ Press **'D'** to add 10 seconds. Press **'#'** to view credits.
- 🌀 The fan activates when the timer reaches zero for **5 seconds**.

## Schematic and Visuals



## 🔩 Component List

| Component | Quantity | Description |
|---|---|---|
| Arduino Uno R3 | 1 | Microcontroller board |
| LCD 16x2 | 1 | Display with I2C interface |
| Keypad 4x4 | 1 | User input for timer |
| H-Bridge Motor Driver | 1 | L298N Motor Controller |
| DC Motor | 1 | Acts as the fan |
| Breadboard | 1 | For easy wiring |

## 🦹 Version Comparison

| Feature | V1 (Basic) | V2 (Enhanced) |
|---|---|---|
| Timer Control | ✅ | ✅ |
| LCD Display | ✅ | ✅ |
| Keypad Interface | ✅ | ✅ |
| Direct Motor Control | ✅ | ❌ |
| H-Bridge Motor Control | ❌ | ✅ |
| Breadboard | ❌ | ✅ |

## ✅ Conclusion

**FanTimer** offers an efficient solution for automated fan control. Its **simple design** and diverse **applications** make it ideal for daily life and beyond! 🌱

## Arduino Code

```
#include <Keypad.h>
#include <Adafruit_LiquidCrystal.h>

const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
Adafruit_LiquidCrystal lcd(0);
int seconds = 0;
bool timerRunning = false;
bool timerPaused = false;
unsigned long previousMillis = 0;
const long interval = 1000;
const int motorPin = 13;
bool showOmarCredits = false;

void resetTimer();
void startTimer();
void pauseTimer();
void add10Seconds();
void enterTime(char key);
void displayTimerValue();
void runMotor(int duration);

void setup() {
  lcd.begin(16, 2);
  lcd.setBacklight(1);
  lcd.print("Enter Time: ");
  pinMode(motorPin, OUTPUT);
}
```

```
void loop() {
  char key = keypad.getKey();
  if (key) {
    if (key == 'A') startTimer();
    else if (key == 'B') pauseTimer();
    else if (key == 'C') resetTimer();
    else if (key == 'D') add10Seconds();
    else if (key == '#') showOmarCredits = true;
    else if (key >= '0' && key <= '9') enterTime(key);
  }
  unsigned long currentMillis = millis();
  if (timerRunning && !timerPaused && currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    if (seconds > 0) {
      seconds--;
      lcd.setCursor(0, 1);
      lcd.print("Time: ");
      lcd.print(seconds);
      lcd.print("s   ");
    } else if (seconds == 0 && timerRunning) {
      timerRunning = false;
      lcd.clear();
      lcd.print("Time's up!");
      runMotor(5000);
      delay(2000);
      lcd.clear();
      lcd.print("Enter Time: ");
    }
  }

  if (showOmarCredits) {
    lcd.clear();
    lcd.print("Was made by");
    lcd.setCursor(0, 1);
    lcd.print("Omar Abdelrahman");
    delay(3000);
    lcd.clear();
    lcd.print("Enter Time: ");
    showOmarCredits = false;
  }
}
void resetTimer() {
  seconds = 0;
  timerRunning = false;
  timerPaused = false;
  lcd.clear();
  lcd.print("Enter Time: ");
}
```

```
void startTimer() {
  if (seconds > 0) {
    timerRunning = true;
    timerPaused = false;
    previousMillis = millis();
  }
}

void pauseTimer() {
  if (timerRunning) timerPaused = !timerPaused;
}

void add10Seconds() {
  seconds += 10;
  lcd.setCursor(0, 1);
  lcd.print(seconds);
}

void enterTime(char key) {
  seconds = seconds * 10 + (key - '0');
  lcd.setCursor(0, 1);
  lcd.print(seconds);
}

void displayTimerValue() {
  lcd.setCursor(0, 1);
  lcd.print("Set: ");
  lcd.print(seconds);
  lcd.print("s ");
}

void runMotor(int duration) {
  digitalWrite(motorPin, HIGH);
  delay(duration);
  digitalWrite(motorPin, LOW);
}
```