**CHAPTER 1**

**INTRODUCTION**

The project "Applications of TM1637 4-Digit 7-Segment Display" explores the implementation and practical uses of the TM1637 display module in various applications. TM1637 is a widely used integrated circuit that allows users to interface with four-digit seven-segment LED displays. These displays are commonly employed in projects where numerical and alphanumeric information needs to be visually presented.

The main objective of this project is to demonstrate how the TM1637 display module can be effectively utilized to showcase different types of data and information. The simplicity and versatility of the TM1637 module make it a popular choice for hobbyists, students, and even professionals in their electronic projects.

A comprehensive introduction to the TM1637 display module, including its features, functionality, and technical specifications. Understanding the capabilities of this module will provide the foundation for exploring its applications. Explanation of the necessary hardware connections, Interfacing and communication protocols required to interface the TM1637 display module with a microcontroller or a development board. We will explore different platforms and programming languages to drive the display effectively.

Throughout the project, we will provide step-by-step guides, sample code, and circuit diagrams to facilitate the implementation of each application. The project's goal is to encourage creativity and innovation while demonstrating the wide range of practical uses for the TM1637 display module in the world of electronics and embedded systems.

**CHAPTER 2**

**COMPONENTS REQUIRED**

a. Arduino UNO

b. TM1637 Display

c. DTH11/DTH22

d. Breadboard

e. Connecting wire.

**COMPONENT DESCRIPTION**

1. **ARDUINO UNO**:

A microcontroller that can connect to electrical components and program them using software. The Arduino Uno has a series of black pins that connect to electrical components via wires. The pins have different functions depending on what they’re labelled: a) power pins, b) digital pins, and c) Analog pins.

a. The power pins provide power and ground to the breadboard. The 3V and the 5V pin are 3 volt and 5-volt power supplies, respectively. The GND pin is the ground. This is where the circuit ends. There are two GND pins on the Arduino Uno. One is with the power pins and the other is with the digital pins. We will be using the GND pin near the digital pins.

b. Digital pins can only have two states, ON or OFF. When a digital pin is on (HIGH or 1), it’s supplying power. When it’s off (LOW or 0), it’s not supplying power. We can control whether the pin is on or off using software. The digital pins are labelled 0 - 13. They can act either as an input, receive sensor information, or output, send instructions or power to a component, in the circuit. We will be using the digital pins as outputs in this project. They will be supplying the LED with power based on our software instructions.

c. Analog pins can have a variety of values beyond just off or on. They can be any fraction between 0 and 1. The Analog pins are labelled A0 - A5.

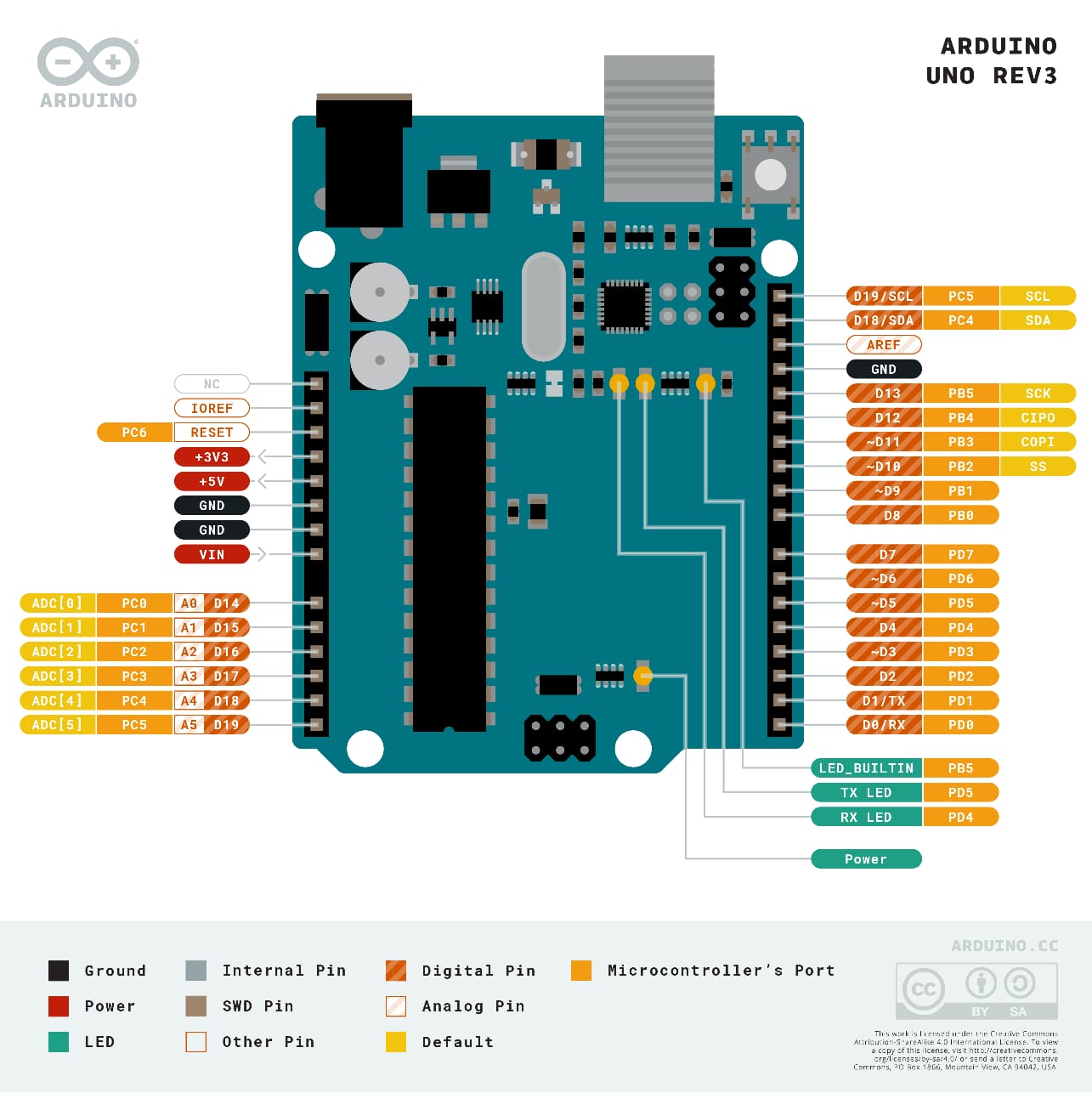


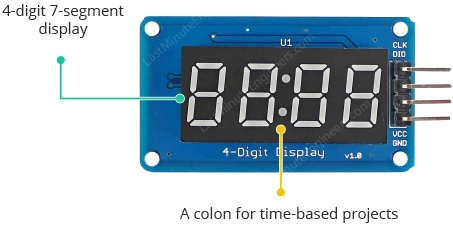
Fig.1 : Arduino Uno

1. **TM1637 DISPLAY:**

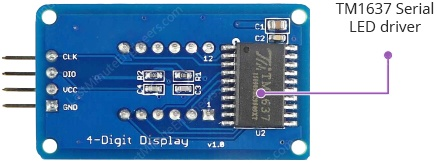
The **TM1637** is a display driver IC that can control up to four 7-segment displays with decimal points. It is commonly used in Arduino projects to display sensor data, temperature, time, etc. [The TM1637 IC mounted on the back of the display module reduces the number of connection pins required from 12 to just four](https://www.makerguides.com/tm1637-arduino-tutorial/). [The TM1637 communicates with the processor in an I2C-like protoco](https://reference.arduino.cc/reference/en/libraries/tm1637/)l.

The implementation is pure software emulation and doesn't make use of any special hardware (other than GPIO pins). It is assumed that pull-up resistors are present (usually integrated in the display module).

The TM1637 module combines a classic 0.36″ 4-digit 7-segment display and the TM1637 LED driver from Titan Micro Electronics, allowing you to control all digits, and the colon using only two I/O pins.



The TM1637 module is ideal for displaying sensor data or temperature. It also includes a “colon” for use in clock and time-related projects. The TM1637 handles all the work of refreshing the display after it has been updated by the microcontroller, which is a nice bonus. This frees up the microcontroller to do other important things.



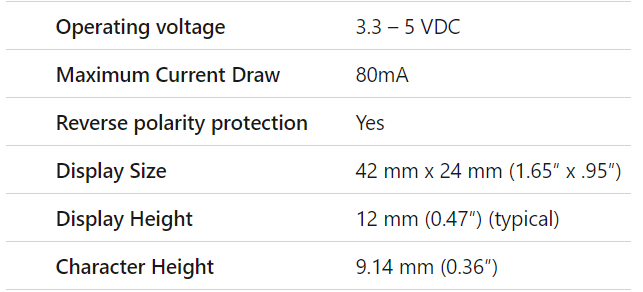


Table1: Technical Specifications

CLK PIN : A clock input pin. Connect to any digital pin on Arduino.

DIO PIN : A Data I/O pin. Connect to any digital pinon Arduino.

VCC PIN : A pin supplies power to the module. Connect it to the 3.3V to 5V power supply.

GND PIN : A ground pin.

1. **DTH22**

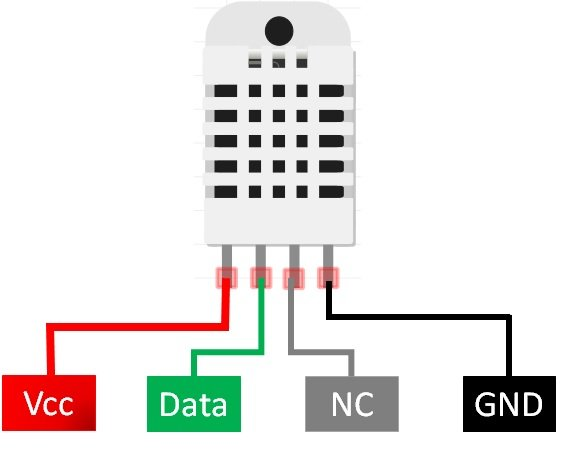
****

Fig.3 : DTH 22

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer. Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enables DHT22 to be suited in all kinds of harsh application occasions. Single row packaged with four pins, making the connection very convenient.

Table 2: Technical specifications

A screenshot of a computer

Description automatically generated

1. Breadboard is required for the construction of the circuit and connecting wires and jumper wires for interconnection of the circuit.

**CHAPTER 3**

**CIRCUIT DIAGRAM**

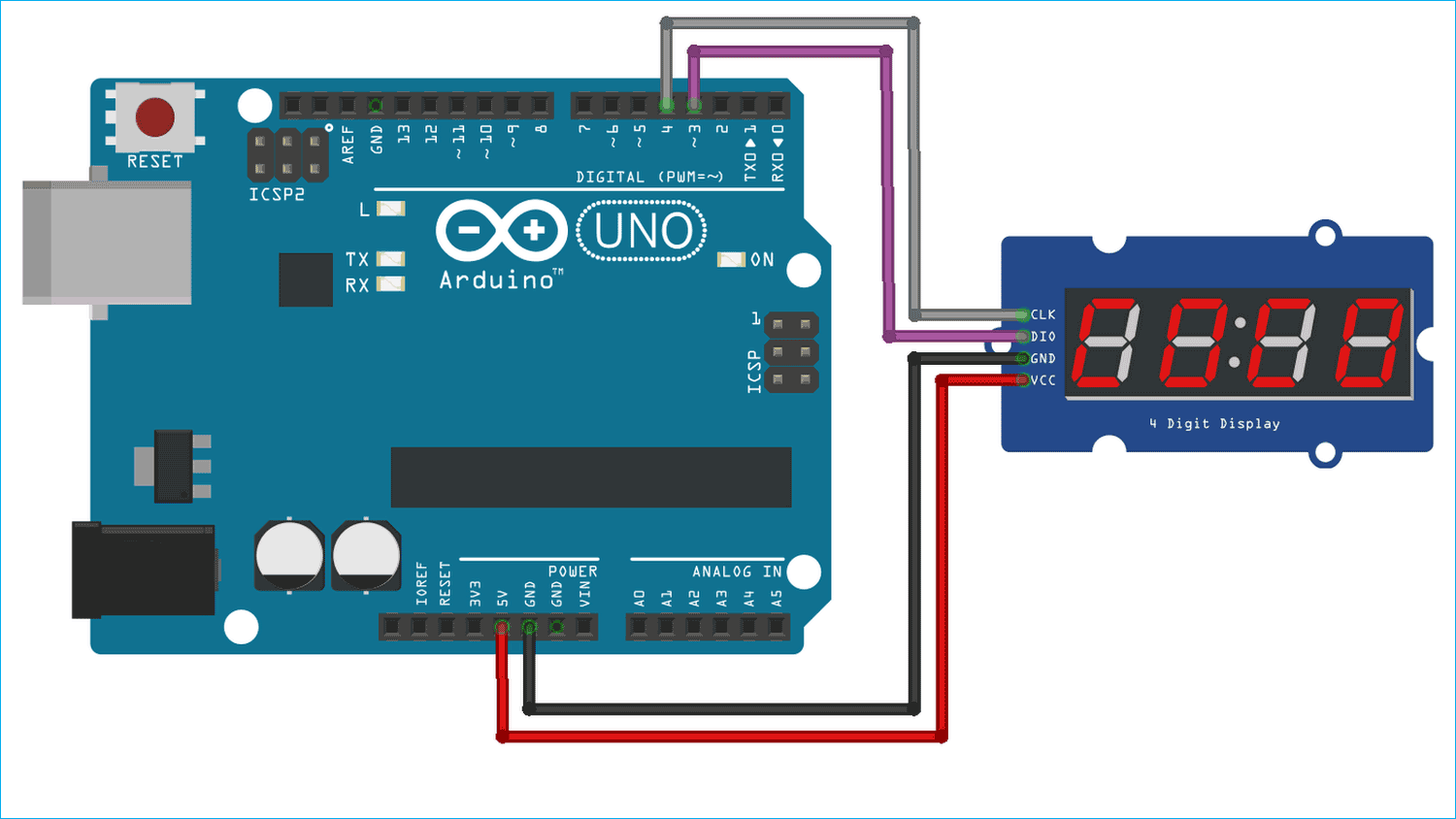


Fig.4 : INTERFACING TM1637 WITH ARDUINO UNO

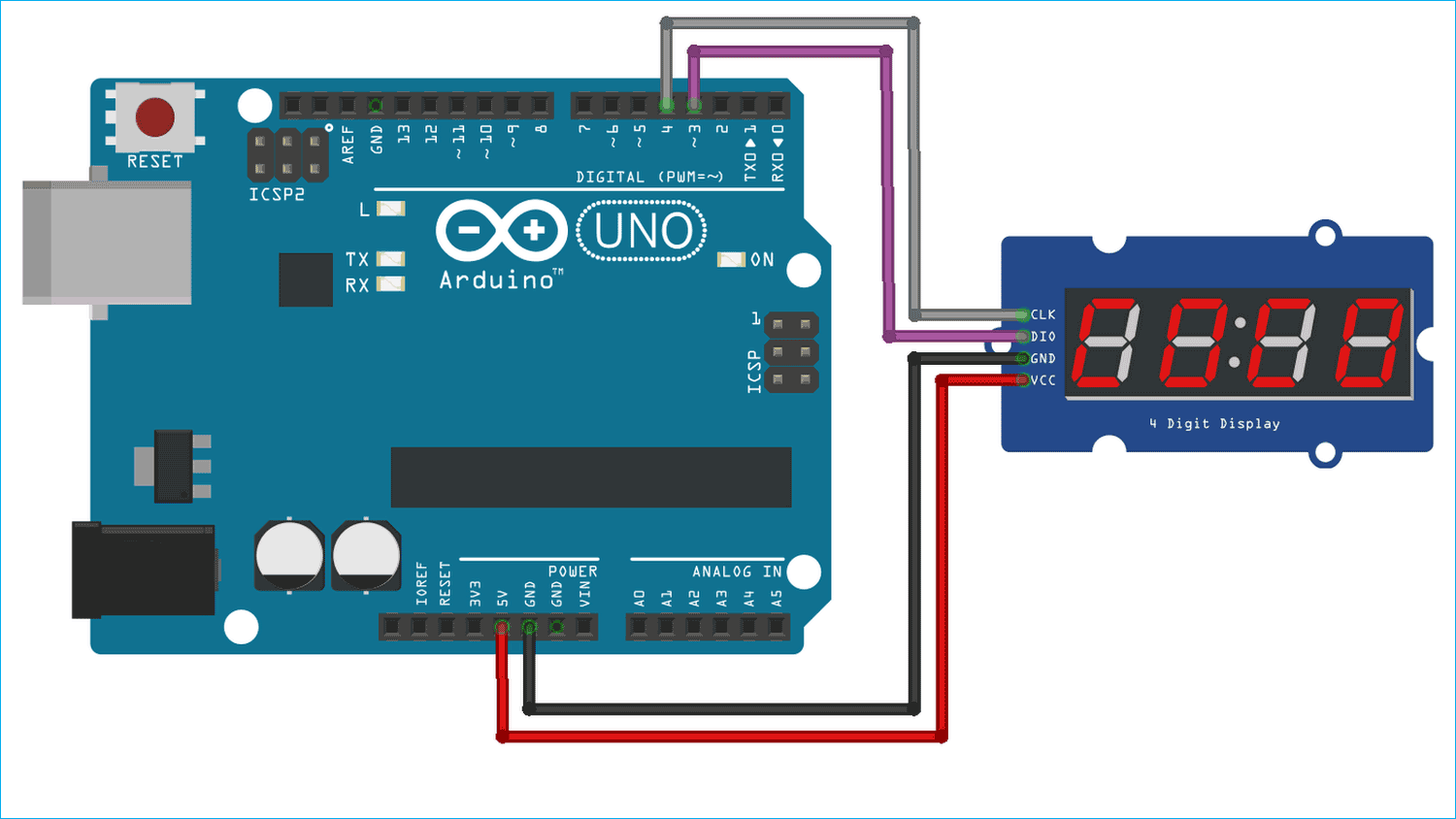
**Step by step procedure:**

1. The above circuit diagram of shows interfacing of TM1637 Display with Arduino Uno.
2. To connect a TM1637 display to an Arduino, connect four wires: Two for power lines (3.3V or 5V pin) and two for controlling the display.
3. The module can be powered from the 3.3V or 5V output of the Arduino. Connect the CLK and DIO pins to any digital pins of Arduino.
4. For example, Pins-3 and 4 on the Arduino. The pin numbers in the code should be changed if different pins are used.
5. We just need to define the pins through which the 7-segment display module is connected to the Arduino and other things will be handled by the [Arduino TMS1637 Library](https://github.com/avishorp/TM1637).
6. Upload the codes to process different applications of the TM1637 Display.

**CHAPTER 4**

**WORKING**

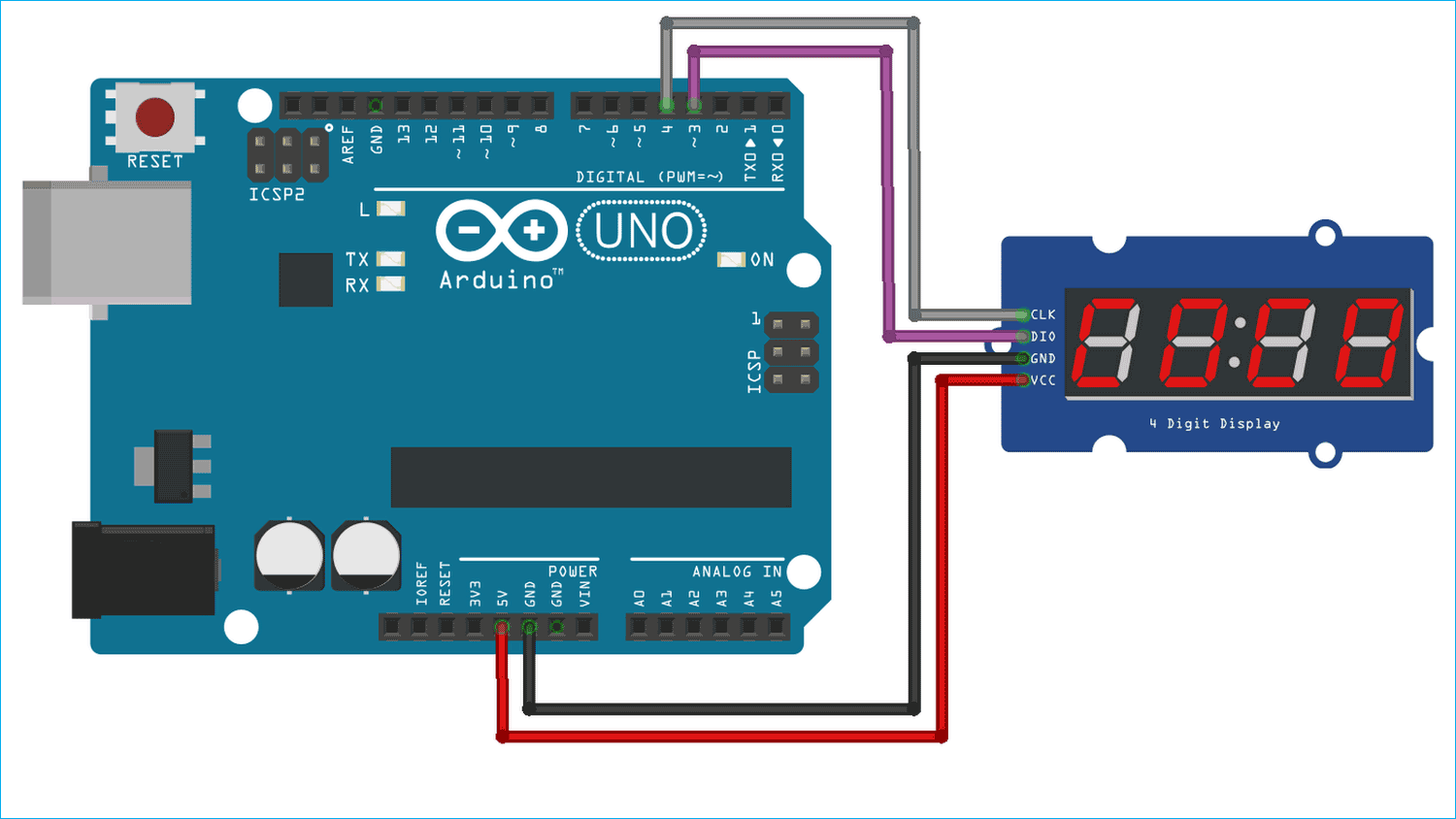
**EXAMPLE 1: DISPLAYING STRING AND A NUMBER**



**Steps for working with the TM1637 display to show a string and a number:**

1. Initialize Communication: Connect the TM1637 module to microcontroller (Arduino, Raspberry Pi, etc.) via the required pins (CLK and DIO). Initialize the communication and configure the display settings.
2. Data Format: Convert the characters of the string and the digits of the number into their corresponding 7-segment display codes. There are predefined codes for numbers 0-9 and some letters.
3. Data Transmission: Send the data for each segment of the display one by one using the CLK (clock) and DIO (data input/output) pins. This involves sending the command to set the display brightness, followed by the segment data.
4. String Display: For displaying a string, we'll need to cycle through each character in the string and convert it to its corresponding 7-segment code. Transmit the segment data for each character one by one to the display.
5. Number Display: For displaying a number, break down the number into its individual digits. Convert each digit to its 7-segment code and send the segment data for each digit.

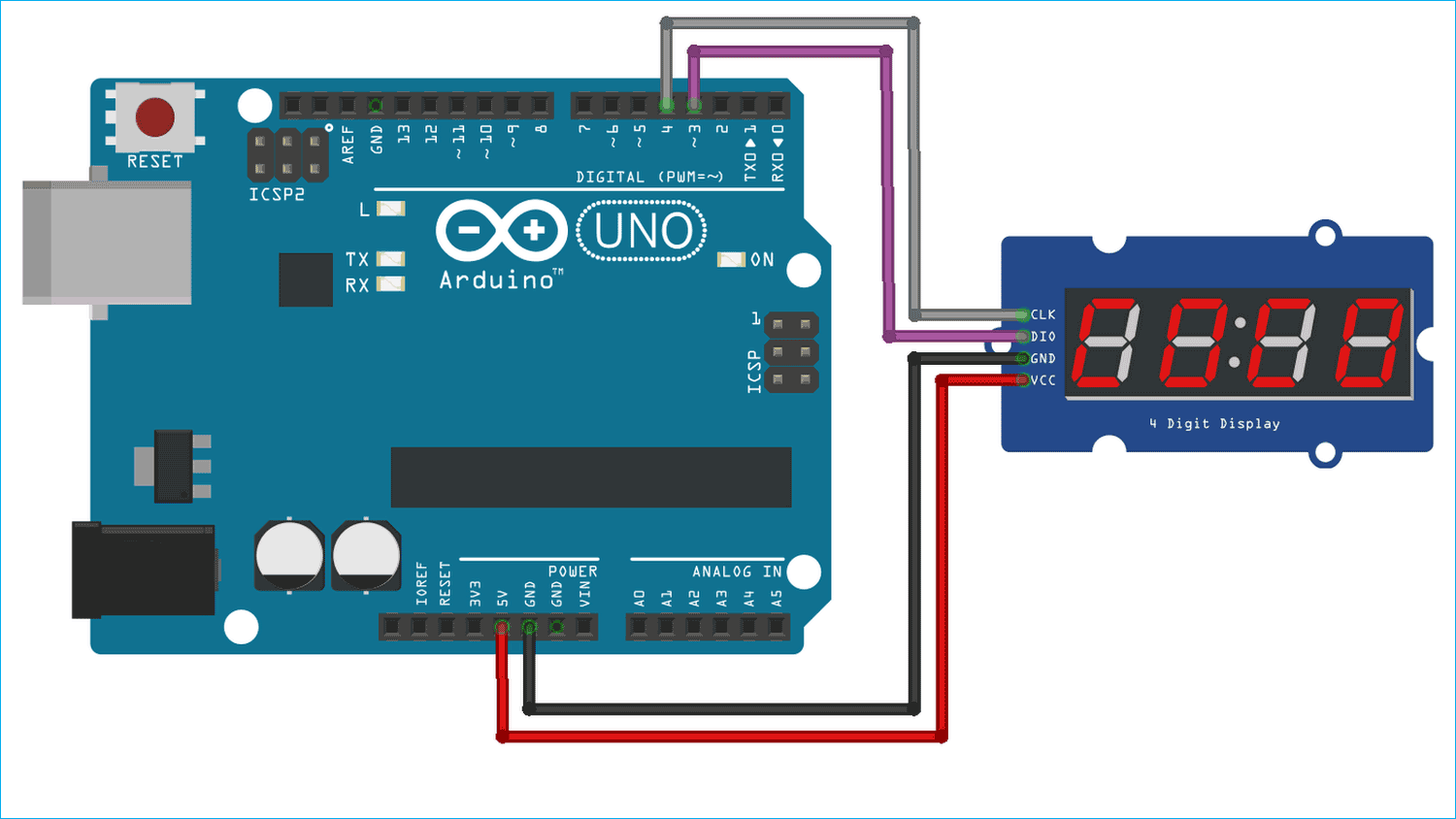
**EXAMPLE 2: DISPLAYING SCROLLING AND BLINKING TEXT**



**Steps for working with the TM1637 display for scrolling and blinking text:**

1. Initialize Communication: Set up the communication between the Arduino and the TM1637 module using the required pins (CLK and DIO). Initialize the display settings, such as brightness and other options.
2. Data Format: Convert the characters of the text you want to display into their corresponding 7-segment display codes. Refer to the TM1637 datasheet or library documentation for the codes.
3. Scrolling Text: To display scrolling text, break down the text into individual characters or segments. Create a loop that cycles through each character, showing them on the display one by one. We can create the scrolling effect by introducing delays between each character display.
4. Text Display: If we want to display static text, we can use a similar approach to scrolling text but without the delay between characters. Simply transmit the segment data for each character of the text to the display.
5. Display Update Frequency: For smooth scrolling, adjust the delay between each character display to achieve the desired scrolling speed. Make sure to update the display at a frequency that maintains readability.
6. Looping: Depending on our desired behavior, we can run the scrolling or text display process in a loop to continuously update the display with the desired content.

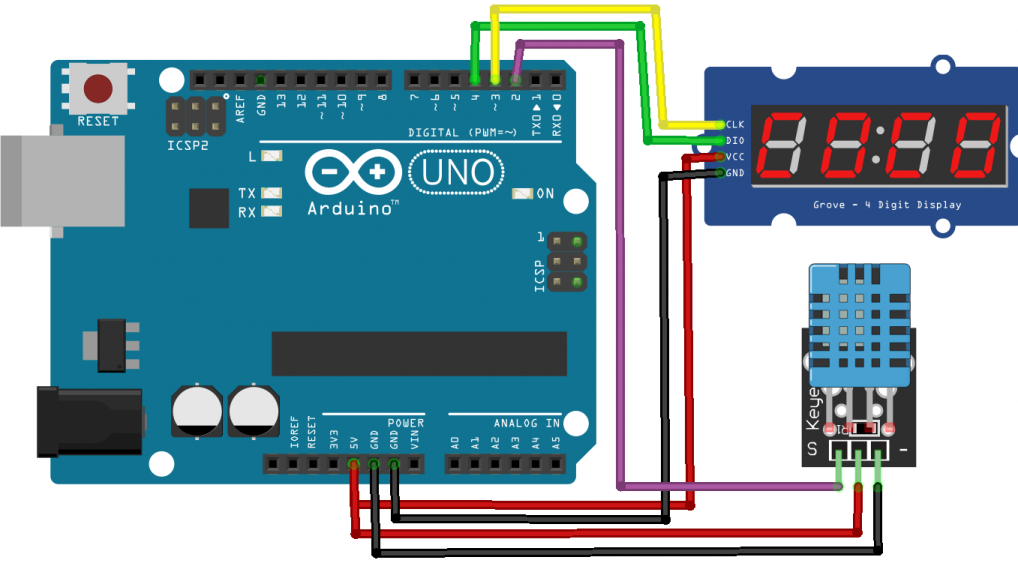
**EXAMPLE 3: Creating 4-Digit Counter**



**Steps for working with the TM1637 display to create a 4-digit counter:**

1. Initialize Communication: Connect the TM1637 module to Arduino via the required pins (CLK and DIO). Initialize the communication and configure the display settings.
2. Data Format: Convert the numbers from 0 to 9999 into their corresponding 7-segment display codes. We'll need to break down each digit into its individual segments.
3. Counter Logic: Set up a loop that runs indefinitely. This loop will act as the counter. Initialize a variable to hold the current count value (starting from 0).
4. Update Display: Inside the loop, convert the current count value into its individual digits (thousands, hundreds, tens, and units).
5. Send Data: Transmit the segment data for each digit to the TM1637 display module using the CLK and DIO pins. Ensure that you send the segments for each digit sequentially.
6. Delay: Introduce a delay between count updates. The delay should be chosen to provide a visible update rate without making the counter too fast or too slow.
7. Increment Counter: After updating the display, increment the count variable. If it reaches 10, reset it to 0 and increment the next digit. Repeat this logic for all digits (tens, hundreds, thousands), considering carry-over as needed.
8. Looping: Keep the loop running to continuously update the counter and display the digits.

**EXAMPLE 3: Displaying Temperature using DTH11/22**



**Steps for working with the TM1637 to show Temperature using DTH11/22:**

1. Initialization: Connect the DHT11/DHT22 sensor and the TM1637 module to your microcontroller. Set up the necessary pins and install the required libraries for both components.
2. Sensor Setup: Initialize the DHT sensor using the library functions. Configure it to read temperature data.
3. Read Temperature: Use the DHT library to read the temperature value from the sensor. This value will be in Celsius.
4. Convert to Fahrenheit: Calculate the Fahrenheit temperature using the formula,

F=(Celsius \* 9/5) + 32.

1. Segment Conversion: Convert both the Celsius and Fahrenheit temperature values into their corresponding 7-segment display codes using the TM1637's segment codes. Break down each digit into its individual segments.
2. Update Display: Transmit the segment data for each digit of the Celsius temperature value to the TM1637 display module using the CLK and DIO pins. Display either Celsius or Fahrenheit temperature based on user preference.
3. Loop and Delay: Set up a loop that continuously reads the temperature, converts it, updates the display, and introduces a delay between updates. This loop ensures that the display stays updated with the latest temperature information.

**CHAPTER 5**

**ADVANTAGES AND DISADVANTAGES**

* 1. Advantages of using TM1637 for Displaying String and a Number:

1. Ease of Implementation: TM1637 simplifies the process of displaying numbers and limited strings on 7-segment displays, making it suitable for applications that require basic numeric and text representation.
2. Readability: 7-segment displays are well-suited for displaying numbers and simple text, ensuring good readability even from a distance.
3. Low Complexity: The TM1637's interface and libraries make it relatively simple to integrate into projects without requiring extensive programming knowledge.
   1. Disadvantages of using TM1637 for Displaying String and a Number:
4. Limited Text Display: While you can display limited text, it's not designed for more extensive textual content or detailed messages.
5. Lack of Graphics: The TM1637 is limited to displaying numbers and basic letters, making it unsuitable for graphical content or symbols.
   1. Advantages of using TM1637 for Displaying Scrolling Text:
6. Basic Animation: TM1637 supports basic animation effects like scrolling text, providing some visual interest to displays.
7. Attention-Grabbing: Scrolling text can capture attention and convey important messages effectively.

2.2 Disadvantages of using TM1637 for Displaying Scrolling Text:

1. Limited Animation: The animation capabilities are limited to basic scrolling, and more complex animations or transitions are not feasible.
2. Limited Text Length: The amount of text that can be scrolled is restricted by the display's physical size and the available segments.

3.1 Advantages of using TM1637 for Creating a 4-Digit Counter:

1. Compact Size: TM1637 allows you to create a compact 4-digit counter display suitable for various applications.
2. Real-Time Updates: The TM1637's simple interface enables real-time updates of the counter's value without much processing overhead.

3.2 Disadvantages of using TM1637 for Creating a 4-Digit Counter:

1. Limited Digits: The 4-digit counter has limitations when displaying large numbers or in scenarios requiring more digits.
2. Limited Interactivity: TM1637 lacks advanced user interaction features that could enhance the counter's functionality.

4.1 Advantages of using TM1637 for Displaying Temperature (C and F) using DHT11/22:

1. Basic Temperature Display: TM1637 is suitable for basic temperature displays without requiring advanced graphical capabilities.
2. Easy Integration: Integrating the temperature readings with the TM1637 is relatively straightforward using libraries.

4.2 Disadvantages of using TM1637 for Displaying Temperature (C and F) using DHT11/22:

1. Limited Sensor Integration: While you can display temperature values, integrating other sensor data or advanced features could be challenging.
2. Precision and Resolution: The limited number of display segments might affect the precision of temperature display, especially when showing decimal values.

**CHAPTER 6**

**APPLICATIONS**

TM1637 Display can be used in various applications with an Arduino for the scenarios mentioned:

1. Displaying String and a Number:

* Application: Counting events or items and displaying a label or description alongside the count.
* Example: Display "Bottles: 123" where "Bottles" is the string and "123" is the count.
* Implementation: Convert the count and string to their respective segment codes, then display them using the TM1637.

2. Displaying Scrolling and Displaying Text:

* Application: Creating simple informational displays or announcements.
* Example: Scroll messages like "Welcome to XYZ Store" or display static messages like "Meeting at 3 PM."
* Implementation: Use a loop to update the display with new characters to create scrolling or static text effects.

3. Creating a 4-Digit Counter:

* Application: Counting occurrences, events, or tracking quantities.
* Example: Display the number of steps taken, items produced, or people in a room.
* Implementation: Increment a counter variable and update the display with the counter's value using the TM1637.

4. Displaying Temperature (°C and F) using DHT11/22:

* Application: Monitoring and displaying room temperature in Celsius and Fahrenheit.
* Example: Display "Temp: 25°C / 77F" to show both Celsius and Fahrenheit temperatures.
* Implementation: Read temperature data from DHT11/DHT22, convert to Celsius and Fahrenheit, and display using the TM1637.

The TM1637, when used with an Arduino, provides a convenient and straightforward way to create these types of applications. It's suitable for scenarios that require basic numeric and textual information to be displayed on a 7-segment display. However, for more complex visualizations, graphical displays, or advanced user interactions, we might need to consider other display solutions with higher capabilities.

**CHAPTER 7**

**FUTURE SCOPE**

Here are some potential future scopes for applications of TM1637 display with Arduino Uno:

* The future scope of using the TM1637 with Arduino encompasses a range of technical advancements and creative applications. Firstly, enhancing user interaction by integrating buttons or sensors can allow users to toggle between Celsius and Fahrenheit modes or control scrolling speeds.
* Additionally, integrating wireless communication modules like Bluetooth or Wi-Fi could facilitate remote monitoring and control of display content. As the IoT ecosystem expands, the TM1637's potential grows, enabling integration with smart home setups for home automation and cloud connectivity.
* Furthermore, the combination of the TM1637 with data logging capabilities could provide insights over time, such as recording temperature or counter data for historical analysis. This data, once collected, might be harnessed for informed decision-making or optimizations.
* In parallel, exploring augmented reality interfaces and AI integration could lead to dynamic content presentation, enhancing both the visual experience and content adaptability. Lastly, advancements in power management could contribute to energy efficiency, automatically adjusting display brightness based on ambient light conditions. As technology continues to evolve, the TM1637's versatility promises to thrive at the intersection of creativity and innovation.

**CHAPTER 8**

**CONCLUSION**

In conclusion, this project involving the integration of the TM1637 display with Arduino showcases the versatility and adaptability of these technologies. By combining the TM1637's simplicity in displaying numbers and text with the capabilities of Arduino, we've successfully created applications ranging from basic numeric displays to more intricate features like scrolling text, temperature monitoring, and interactive counters.

This project not only highlights the ease of implementation and resource efficiency of the TM1637 but also opens the door to future possibilities. Looking ahead the project's future scope holds promising avenues. The integration of advanced user interactions, wireless communication, and cloud connectivity could elevate the practicality and functionality of the TM1637-based displays.

Furthermore, the synergy with data logging, augmented reality, and AI integration has the potential to offer deeper insights and immersive experiences. As we continue to push the boundaries of innovation, the TM1637 and Arduino remain valuable tools that empower creators to bridge simplicity and complexity, shaping a dynamic landscape of applications in the realm of display technology.

**REFERENCES**

[1] Avishay Orpaz's TM1637 Library: -GitHub Repository:

[https://github.com/avishorp/TM1637] (https://github.com/avishorp/TM1637)

[2] SevenSegmentTM1637 Arduino Library by Bram Harmsen: - GitHub Repository:

[https://github.com/bremme/arduino-tm1637] (https://github.com/bremme/arduino-tm1637)

[3] Arduino Playground-TM1637: -Website:

(https://playground.arduino.cc/Main/TM1637/)

[4] MCI Electronics Datasheet for TM1637: - Datasheet:

[https://www.mcielectronics.cl/website\_MCI/static/documents/Datasheet\_TM1637.pdf]

(https://www.mcielectronics.cl/website\_MCI/static/documents/Datasheet\_TM1637.pdf)

[5] TM16xx Library for TM16xx Chip Family: - GitHub Repository:

[https://github.com/maxint-rd/TM16xx] (https://github.com/maxint-rd/TM16xx)