An Arduino Word Clock

Tony Jiang, Winchester College, Winchester, Hampshire, SO23 9NA

1. Introduction

This project was a personal initiative that I undertook in my spare time between January and July 2025. At my school, there was a gifted QLOCKTWO EARTH 45 (a designer word clock) mounted on the wall. However, it had been non-functional for over five years. Only a few staff members recalled ever seeing it in operation; for most, it appeared to be nothing more than a decorative object. I discovered that the cost of professional repair was prohibitively high, to the extent that purchasing a new unit would be more economical. This realization motivated me to design and build my own word clock, incorporating custom features and functionality, all at a fraction of the commercial price. My goal was to install my fully functioning version at the school as a practical and symbolic replacement.

Figure 1.1 shows a QLOCKTWO EARTH 45 STAINLESS STEEL, the same model present at my school.[1]

2. The Problem to be Solved

This project aims to design and manufacture a 45 × 45 cm word clock that not only displays the time but also incorporates a range of custom features. These include the ability to play classic games such as Snake and Tetris on its LED matrix, as well as the capability to show scrolling text and dynamic LED effects for aesthetic purposes. Achieving this functionality requires precise control of each individual LED within the matrix, along with a reliable timekeeping system that can maintain accuracy even when the device is disconnected from a power source. To enhance usability and flexibility, the clock is designed to operate on both battery and mains power. Interactive features are enabled through the integration of appropriate input sensors.

In addition to functionality, aesthetic considerations play an important role in the design. The target thickness for the final product is less than 25 mm, closely aligning with the original QLOCKTWO design, which has a main body thickness of approximately 20 mm. It is worth noting that the QLOCKTWO also includes a secondary, smaller rear housing — presumably containing the microprocessor and other electronics — which may allow for a slimmer front profile. This project aims to achieve a similarly sleek form factor while housing all required components internally.

3. Research

3.1 LED Control

Controlling each individual LED in the matrix is a central technical challenge in this project. Through online research, I found that a common approach involves configuring the matrix so that the rows act as anodes and the columns as cathodes. This setup enables the use of multiplexing — a technique where only one row (or column) is activated at a time, while rapidly cycling through all rows (or columns) fast enough that, due to persistence of vision, it appears as though all desired LEDs are lit simultaneously.

The original QLOCKTWO uses a 10×11 LED matrix to accommodate all the necessary words. In my design, I have employed an 11×11 LED matrix to create a more symmetrical display and to provide additional space for customizing the letters. However, this larger matrix requires 22 I/O ports for direct control — 11 for the rows and 11 for the columns. The Arduino Nano unfortunately offers only 14 digital I/O pins, many of which are reserved for other functions. To address this limitation, I will use shift registers such as the 74HC595 to expand the number of available outputs. Each 74HC595 shift register controls 8 outputs while requiring only 3 Arduino pins (data, clock, and latch). However, an additional I/O pin is needed to control the shift registers' clear (reset) line. By daisy-chaining two pairs of shift registers — one pair dedicated to the rows and the other to the columns — I will be able to control up to 32 outputs using just 7 Arduino pins. This configuration enables efficient and scalable management of the 121 LEDs in the matrix while conserving valuable I/O resources on the microcontroller.

The word clock also includes five additional LEDs: one for illuminating the logo and four at the corners to indicate the minute modulus (as the matrix displays time in 5-minute intervals). Since I have unused outputs on the shift registers, I connected the corner LEDs to these outputs. I also decided that the logo LED should be wired directly to the microcontroller to simplify the coding.

3.2 Timekeeping

To ensure accurate timekeeping, I chose to use a separate RTC (Real-Time Clock) module with its own battery backup, as recommended by online research. The module communicates with the microcontroller via I^2C and maintains accurate time even when the main power is disconnected. I chose the AZ-Delivery DS3231, as it was readily available on Amazon.

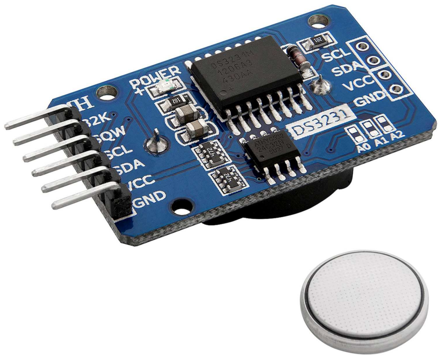

Figure 3.1 shows the AZ-Delivery DS3231 Real-Time Clock (RTC) module, with its 3.3V battery.[2]

3.3 Input Sensor

To enable gameplay features such as *Snake* and *Tetris*, I will need four input buttons positioned near the edges of the word clock. For aesthetic reasons, I want these buttons to remain hidden from view. Recalling that some lamps use touch-sensitive controls, I researched ways to implement invisible inputs and quickly discovered the presence of capacitive touch sensors.

Capacitive touch sensors detect changes in capacitance caused by a conductive object. When a finger nears the surface, it alters the local electric field, which the sensor detects and registers as a touch input. Looking for ways to implement this in my project, I found the Adafruit AT42QT1070 5-Pad Capacitive Touch Sensor. This standalone module communicates with the microcontroller via parallel output, pulling the corresponding channel LOW upon detection of a touch, making it straightforward to integrate into my design. I simply need to attach copper plates to the reverse side of the surface where input is desired and connect the plates to the sensor chip.

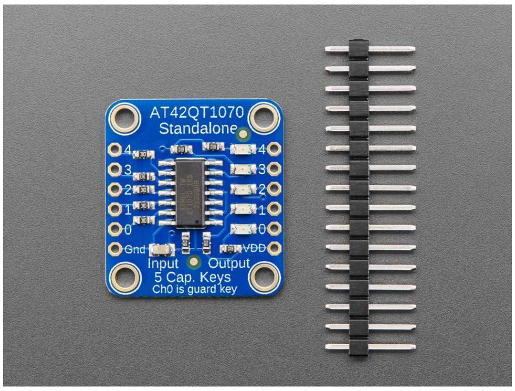

Figure 3.2 shows the Adafruit AT42QT1070 Standalone 5-Pad Capacitive Touch Sensor.[3]

3.4 Power

Since the 126-LED word clock must operate on both battery and mains power, a dedicated voltage regulator separate from the Arduino Nano is necessary. The Arduino cannot directly power such a large number of LEDs. I selected the Pololu 5V 1A Step-Up/Step-Down Voltage Regulator (S13V10F5), which provides a stable 5V output from input voltages ranging between 3V and 22V. For the battery, I plan to use four AAA cells in series. To protect the circuit from damage caused by excessive battery drain, a voltage supervisor is required to disconnect power when voltage falls below a safe threshold. I chose the Texas Instruments TPS3710 IC for this function. The TPS3710 outputs a LOW signal when the input voltage falls below a pre-set threshold. When integrated into a custom circuit, this output can be used to disconnect the circuit.

4. Design and Build

4.1 Schematic

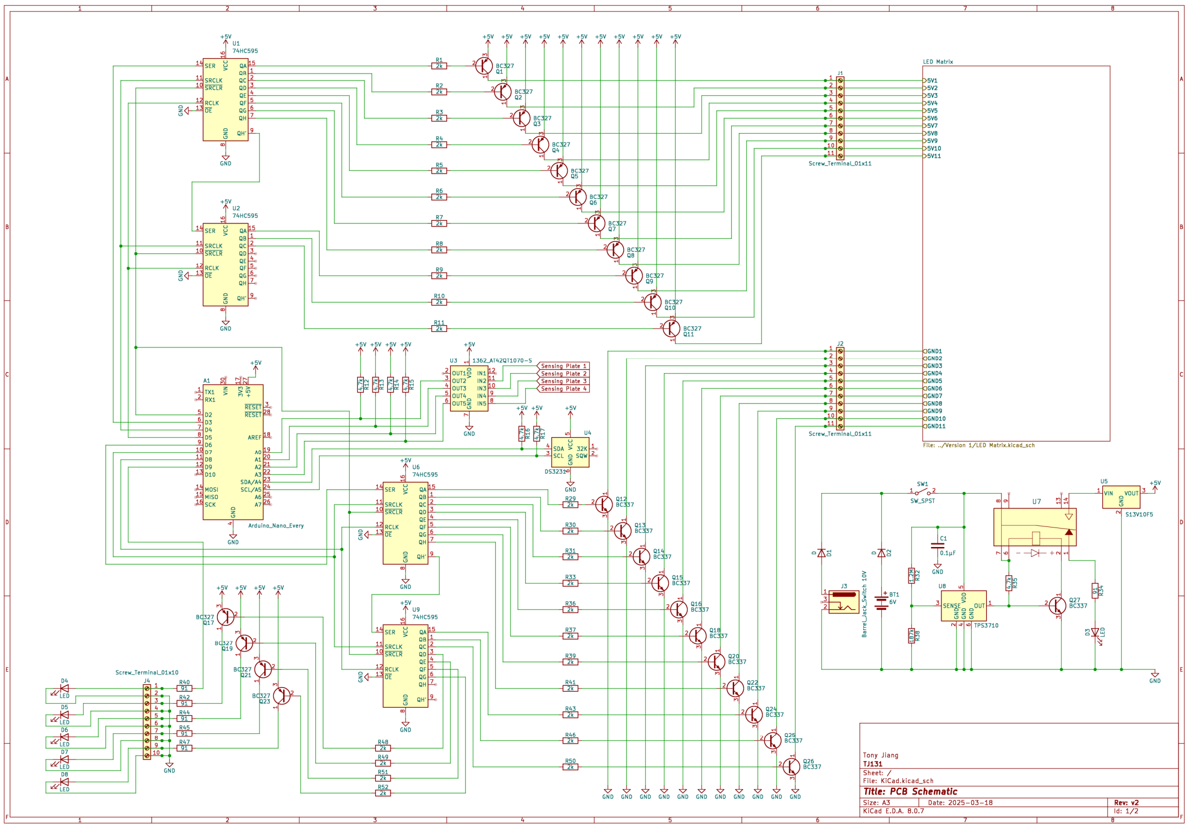

Figure 4.1 shows the schematic of the entire word clock.

Each LED in the matrix is arranged such that all anodes in a given row are soldered together, and all cathodes in a given column are connected. A PNP transistor is placed before each row to switch the row on or off, while an NPN transistor is placed after each column to switch the column on or off. This configuration allows control of any individual LED by activating its corresponding row and column transistors. It also enables each LED to draw current directly from the power source, while still being controlled via the Arduino.

Although the LEDs are rated for 3V, they are powered at 5V because multiplexing across the 11×11 matrix limits the time each LED is active, effectively reducing the average current through each one by a factor of 11. All LEDs are therefore wired in parallel to the 5V line.

Each transistor base is controlled by a 74HC595 shift register, which converts the Arduino's serial output into parallel control signals. A 2 kΩ resistor is placed between each shift register output and transistor base to limit base current and protect the components.

A similar PNP-based switching method is used to control the four corner LEDs, which indicate minutes between the five-minute intervals. However, since these LEDs are not multiplexed, a 91 Ω resistor is added to each to keep the current within safe limits. The logo LED is connected directly to an Arduino I/O pin, also with a 91 Ω resistor for current limiting.

The RTC module (DS3231) requires two additional ports — Serial Data (SDA) and Serial Clocl (SCL) — for communication via the I^2C protocol. I^2C is an open-drain communication protocol where devices can only pull the line LOW; HIGH signals are achieved through external pull-up resistors, which ensure the lines return to a HIGH state when not being pulled LOW. Although the DS3231 module includes built-in pull-up resistors, I added external ones in my circuit as an extra precaution to ensure signal reliability.

The remaining Arduino I/O ports are connected to the Adafruit AT42QT1070 capacitive touch sensor. Although the chip supports five inputs, I only required four sensor plates — one for each invisible button — so I connected four of its output lines to the Arduino. As mentioned earlier, the sensor uses open-drain outputs: it pulls the corresponding line LOW when a touch is detected. To ensure the lines return to a HIGH state when not pulled LOW, I added external pull-up resistors to each connection.

In total, I used 14 signal pins on the Arduino: 8 for the LEDs, 2 for the RTC module, and 4 for the capacitive touch sensor.

The final part of the circuit is the power module, located in the bottom corner of Figure 4.1. The S13V10F5 regulator converts any input voltage between 3V and 22V into a stable 5V output. A reed relay (U7) is used to disconnect the power supply from the main circuit if the input voltage drops too low. This cutoff mechanism is controlled by the TPS3710 voltage monitor, which constantly measures the voltage at its SENSE pin. If the sensed voltage falls below 0.4V, the TPS3710 pulls its open-drain OUT pin LOW. To ensure the OUT pin returns to a HIGH state when not asserted, a pull-up resistor is required.

Since the TPS3710 only detects whether the voltage is above or below 0.4V, a potential divider is used to scale the input voltage accordingly. By selecting resistor values that make the SENSE pin read exactly 0.4V when the input voltage reaches the desired cutoff threshold, the TPS3710 can effectively monitor supply levels. The appropriate resistor values were calculated using equations from the Texas Instruments application report[4] and the TPS3710 datasheet[5], rounded to the nearest standard resistor values.

$$R_2 = \frac{V_{IT}\frac{|\%ACC|}{100}}{|I_S|\left(\frac{V_{IT}}{V_{REF}}-1\right)} = \frac{3.00\text{V}\frac{1}{100}}{25\text{nA}\left(\frac{3.00\text{V}}{0.400\text{V}}-1\right)} = 185\text{k}\Omega \tag{4.1}$$

$$R_1 = R_2\left(\frac{V_{IT}}{V_{REF}}-1\right) = 185\text{k}\Omega\left(\frac{3.00\text{V}}{0.400\text{V}}-1\right) = 1.2\text{M}\Omega \tag{4.2}$$

When used in conjunction with an NPN transistor, this circuit cuts off current to the relay coil if the input voltage is too low, because when the OUT pin is asserted (pulled LOW), the transistor is switched off, preventing current from flowing through the relay magnet. This causes the reed switch to disconnect the main circuit and instead complete a path to an LED, which lights up to indicate that the input voltage has fallen below the threshold — either due to low mains voltage or a depleted battery.

4.2 PCB Layout

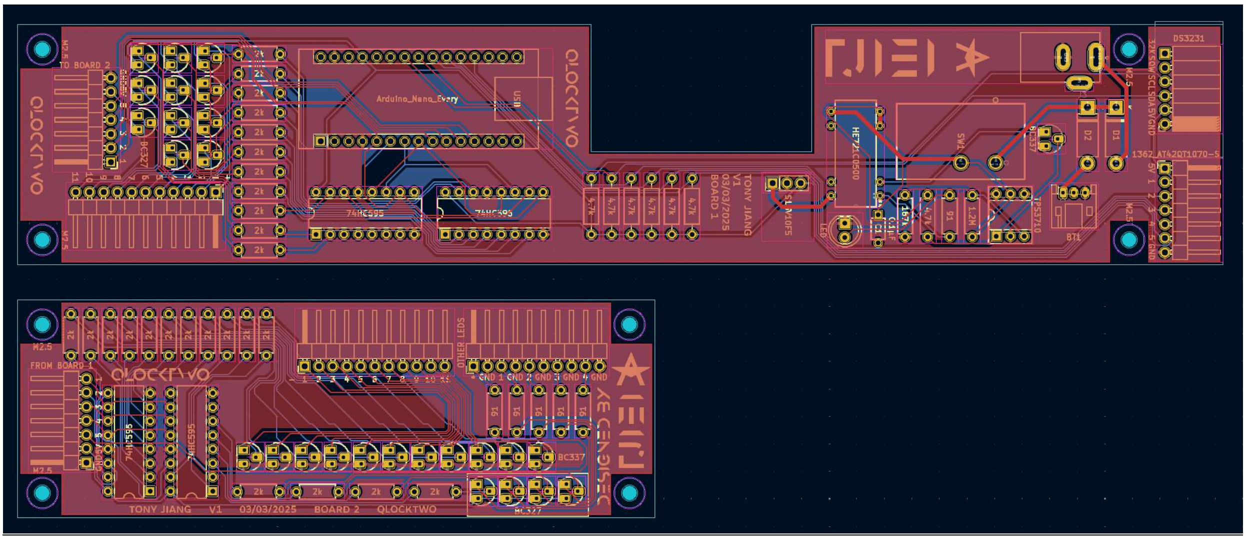

Figure 4.2 illustrates the layout of the custom PCB manufactured for the word clock.

The components from the schematic are distributed across two custom-designed PCB boards. The main challenge in the PCB layout was fitting all components within the limited internal space of the word clock, which has a maximum thickness of 25 mm. The central area is occupied by the LED matrix, leaving only narrow strips along the four edges for electronics. Additionally, the presence of corner LEDs and capacitive touch pads further reduces the space available to mount the PCB boards. On Board 1, there is a central cutout in the PCB to accommodate a copper plate acting as a touch sensor, located at the midpoint of each edge. Various pin headers connect the PCB to the rest of the components:

- On Board 1, the two right-side pin header rows connect the Arduino to the RTC module and capacitive touch sensor.
- The header orientated downwards on Board 1 connects to the anodes of the LED matrix.
- The left-side headers on both boards are wired together to allow communication between the boards.
- On Board 2, the headers oriented upwards connect to the cathodes of the LED matrix and the five additional LEDs (four minute indicators and the logo LED).

The PCB Gerber files were sent to JLCPCB for physical fabrication, which arrived by post within a week.

4.3 Physical Fabrication

The physical fabrication of the product primarily involved laser cutting, supplemented by some woodworking for structural components.

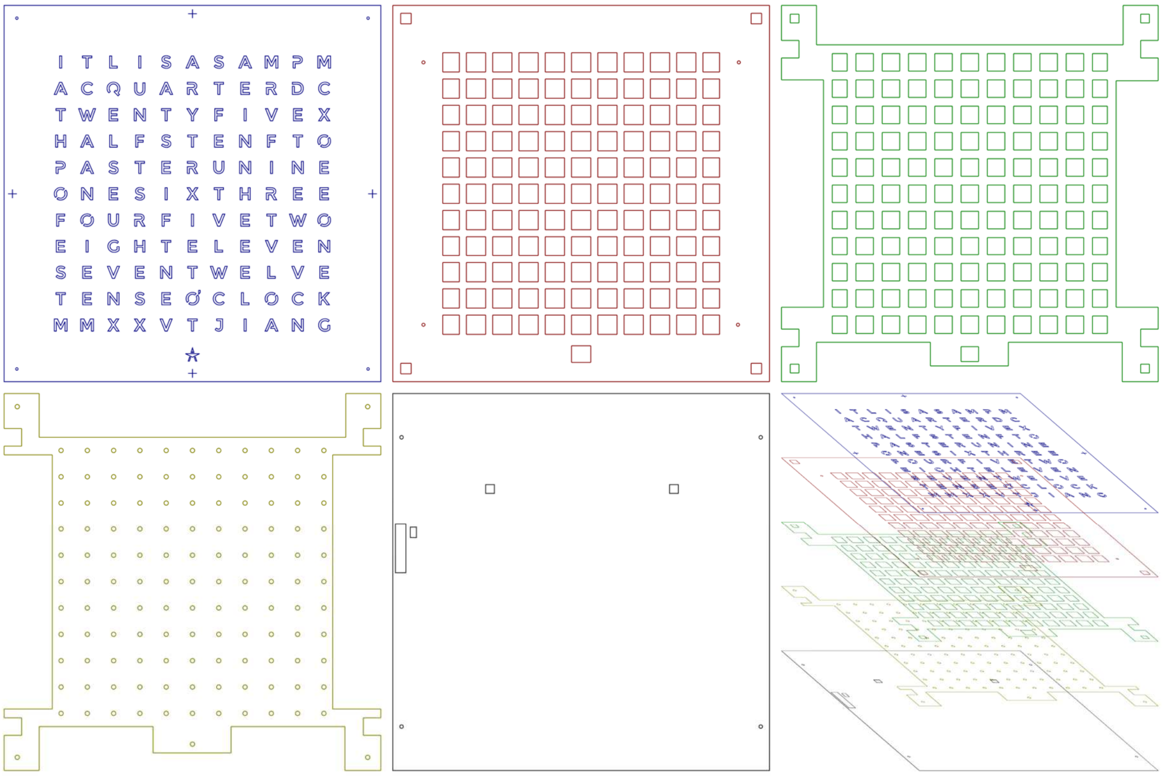

Figure 4.3 illustrates the various structural layers of the word clock. The clock consists of a 0.5 mm vinyl faceplate (blue), 3 mm acrylic light-diffuser layer (red), 9 mm MDF structural layer (green), 3mm acrylic LED structure layer (yellow), and a 3mm back plate (black).

Figure 4.3 shows all the laser-cut layers used in the construction of the word clock. The top blue layer is a sheet of black woodgrain-patterned vinyl with cut-out letters. The cut-out letters are based on the Montserrat font, which I customized to ensure no islands appear within the letters.

The blue layer is adhered to the red layer beneath it, which is made from 3 mm black acrylic with letter-shaped cutouts filled with 3 mm semi-transparent white acrylic. This combination acts as a diffuser, helping to evenly distribute the light from the LEDs below each letter.

The green layer is a 9 mm MDF structural component. It creates the necessary spacing between the diffuser and the LED matrix and provides mounting points for both the red and yellow layers via screws.

The yellow layer is a 3 mm acrylic jig that accurately holds each LED in place within its designated hole.

At the back, the black layer is a 3 mm clear acrylic panel that protects the internal electronics. It includes holes for wall-mounting screws and a cutout on the left-hand side for a switch.

Additionally, four wooden support blocks are glued to the red layer and are slotted into cutouts in the green and yellow layers. These blocks allow the back panel to be securely screwed in while maintaining an air gap between the yellow and black layers — providing essential space for wiring and other electronics.

Lastly, a 10 mm thick ash wood frame is built around the clock, joined at the corners with mitre joints. This frame protects the entire structure from the sides and gives it a clean, finished look. The electronic components are mounted in the narrow spaces along the edges of the clock. Wires are soldered to connect the PCBs to the LEDs and copper touch plates. To keep the wiring secure and organized, hot glue is used to fix the wires in place. The internal layout of the electronics is shown in Figure 4.4, which is a backside view of the finished product.

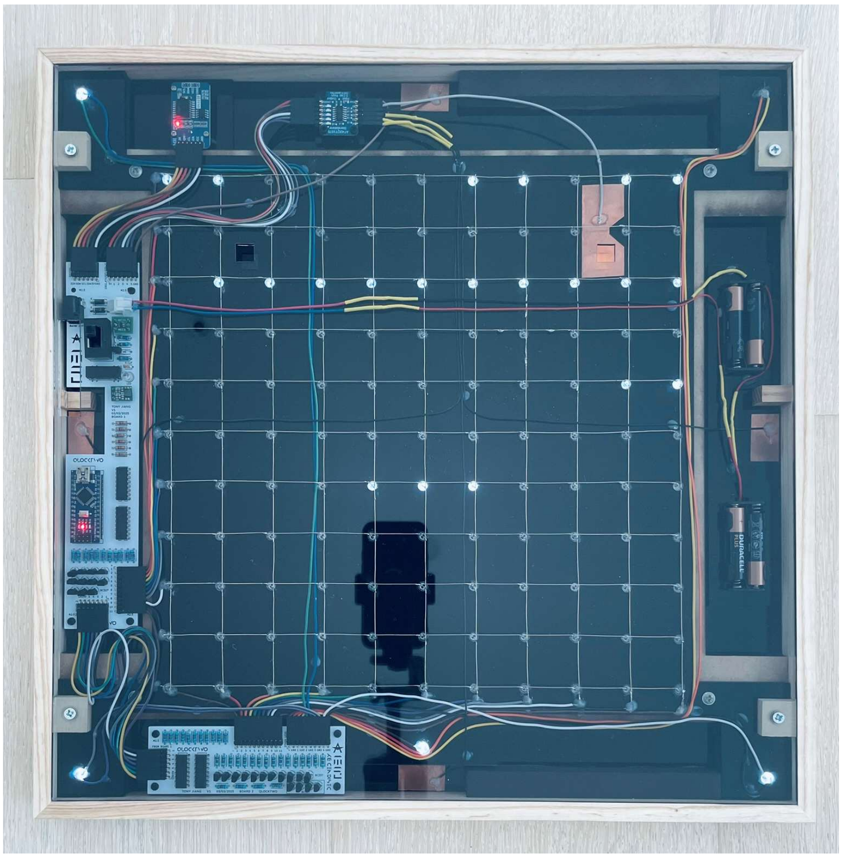

Figure 4.4 shows the rear side of the completed word clock while in clock mode.

<u>4.4 Code</u>

Unfortunately, the full C++ code for this project exceeds 1,000 lines and cannot be fully explained within this document. However, the complete commented code — along with all CAD and design files — is available on the project's GitHub page: https://github.com/TJ131V1/Wordclock.

In short, the code supports four main functions: time display, *Snake*, *Tetris*, and various LED effects. Long pressing the respective copper touch plates switches between modes. The same plates serve as control inputs for *Snake* and *Tetris*. In LED effects mode, users can toggle between several animations: *Conway's Game of Life*, a Matrix-style rain effect, a random letter decay, and a scrolling text display (the text can be modified in the code).

5. Evaluation of Final Product

Figure 5.1 shows the front of the completed word clock, displaying the time 8:47 in clock mode.

The final product has a total thickness of exactly 25 mm, precisely meeting the original design specification. The front display is clean and aesthetically pleasing, and the timekeeping is highly accurate — only about one second behind the actual time. The capacitive touch buttons are extremely well hidden, making them virtually undetectable without prior knowledge of their placement. Gameplay for Snake and Tetris is smooth and responsive, offering a satisfying user experience. The various LED effects add dynamic visual interest, making the clock an eye-catching decorative piece when mounted. The ability to switch modes using capacitive touch sensors enhances usability and creates a seamless, intuitive interface. Overall, I'm very satisfied with the additional features beyond timekeeping — they make the clock both functional and engaging.

The PCB components cost approximately £40. Fabricating the PCB board cost £25 for a minimum batch of five, so in commercial production, the cost per clock would be under £5. The structural components were free, as I fabricated them at my school, but the materials used — acrylic sheets, MDF, and ash dowel — would likely cost under £20. Laser cutting is estimated at around £150 total. Altogether, this brings the total cost per clock to under £200, which is roughly 13% of the price of a QLOCKTWO EARTH 45.

6. Recommendations for Future Work

In hindsight, using multiplexing as the core architecture was not the best design choice. A chain of addressable LEDs would have greatly simplified both the schematic and PCB layout by eliminating the need for shift registers and numerous transistors. Although more expensive than standard white LEDs, addressable LEDs offer major advantages: full RGB colour control, much higher brightness, and more consistent illumination. In the current clock, the centres of the letters are noticeably brighter than the edges, and the display is difficult to read in bright daylight — issues that brighter LEDs would solve. They would also drastically reduce code complexity. The primary reason for the lengthy code is the complexity introduced by the multiplexing architecture.

I also underestimated the small size of the TPS3710's SOT-6 package (just 2.8 × 2.9 mm), which made soldering difficult and time-consuming. For future revisions or commercial production, I would use a voltage supervisor in a larger DIP package.

Additionally, the clock should adjust for daylight saving time automatically, using date information from the RTC to stay accurate year-round in the UK.

Finally, the four AAA batteries only last around two hours. Switching to AA batteries would greatly improve runtime.

7. Acknowledgments

I would like to express my gratitude to Winchester College for providing access to the Design and Technology facilities. I am also sincerely thankful to the D&T teachers and technicians — Callum Barnes, Adrian Ahmed, Trevor Ward, and David Salter — for their invaluable support and guidance throughout the physical fabrication of this project.

8. References

1. QLOCKTWO Earth 45 Stainless Steel Wall Clock Watch QT4ENSS Watch | Jura Watches. https://www.jurawatches.co.uk/products/qlocktwo-classic-stainless-steel-wall-clock-45cm-fcenss.
2. Real Time Clock RTC DS3231 I2C Echtzeituhr kompatibel mit Arduino und Raspberry Pi. *AZ-Delivery* https://www.az-delivery.uk/products/ds3231-real-time-clock.
3. Adafruit Industries Standalone 5-Pad Capacitive Touch Sensor Breakout - AT42QT1070. https://www.adafruit.com/product/1362.
4. Fagnani, A. IQ vs Accuracy Tradeoff In Designing Resistor Divider Input To A Voltage Supervisor. (2011).
5. TPS3710 Wide VIN Voltage Detector datasheet (Rev. A). (2024).