Prezentacja Projektu

Zegar na lampach Nixie

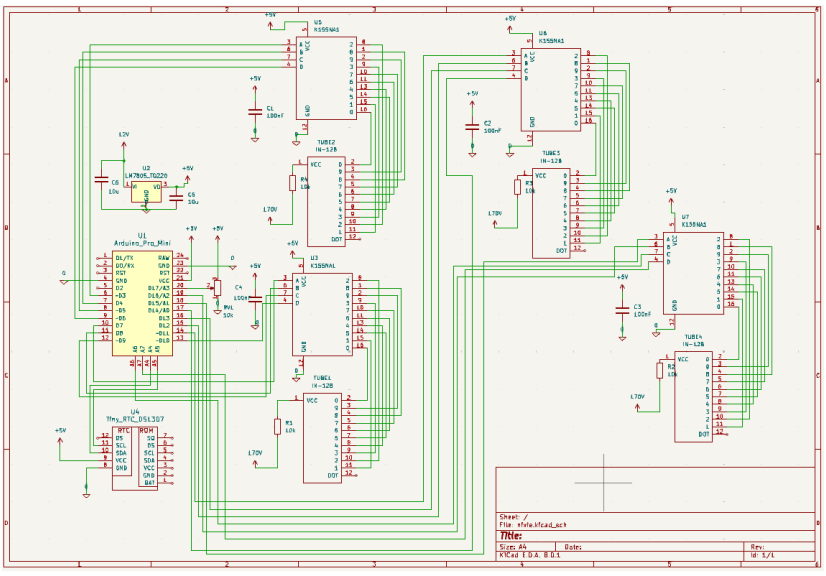
Pavlo Kostushevych

1. Założenia projektowe

- Zegar będzie wyświetlał aktualny czas w formacie godzin i minut za pomocą lamp Nixie.
- Dokładność pomiaru czasu zapewni zastosowanie modułu czasu rzeczywistego (RTC).
- Urządzenie będzie zasilane napięciem stałym.
- Główną jednostką sterującą systemu będzie mikrokontroler **Arduino Pro Mini**, odpowiedzialny za:
  - odczyt danych z modułu RTC,
  - przetwarzanie informacji,
  - sterowanie lampami Nixie.
- Projekt zostanie zrealizowany na dwustronnej uniwersalnej płytce drukowanej, co umożliwi elastyczne rozmieszczenie komponentów oraz ułatwi montaż i lutowanie.
- Do łączenia poszczególnych elementów układu zostaną użyte przewody oraz złącza typu pin-header (męskie i żeńskie), co pozwoli na łatwy montaż, demontaż i modyfikacje w trakcie prototypowania.
- Obwody wysokiego napięcia niezbędne do zasilania lamp Nixie będą odpowiednio zaizolowane i zabezpieczone w celu zapewnienia bezpieczeństwa użytkowania.

2. Schemat (KiCad)

Poniżej przedstawiono schemat ideowy zegara Nixie, opracowany w programie **KiCad**. Projekt podzielono na cztery główne bloki funkcjonalne: zasilanie, mikrokontroler, układ wyświetlania oraz zasilanie wysokiego napięcia.



Schemat ideowy zegara Nixie (opracowany w KiCad)

2.1 Zasilanie

LM7805 pełni rolę stabilizatora napięcia, przekształcając napięcie 12 V na stabilne 5 V zasilające część cyfrową układu. Kondensatory filtrujące eliminują zakłócenia oraz zapewniają stabilność pracy.

2.2 Mikrokontroler

Centralną jednostką sterującą jest **Arduino Pro Mini**. Komunikuje się ono z modułem **RTC DS1307** przez interfejs I²C i wysyła sygnały sterujące do sterowników lamp.

2.3 Wyświetlacze Nixie

Cztery lampy **IN-17** są sterowane przez układy **K155ID1**. Każdy sterownik obsługuje jedną lampę, a połączenia z Arduino umożliwiają wybór aktywnej cyfry. Dodatkowo zastosowano rezystory 10 kΩ do ograniczenia prądu w torze sygnałowym.

2.4 Zasilanie wysokiego napięcia

Lampy wymagają napięcia ok. 170 V DC, które dostarcza moduł **NCH6100HV**. Zasilanie jest odpowiednio zaizolowane na poziomie PCB, co zwiększa bezpieczeństwo użytkowania.

3. Komponenty

Poniżej przedstawiono zestaw elementów wykorzystanych w projekcie zegara Nixie wraz z krótkimi opisami ich funkcji.

Nr	Nazwa komponentu	Symbol / Model	Ilość	Uwagi
1	Mikrokontroler	Arduino Pro Mini	1	5V, 16 MHz
2	Moduł zegara czasu rzeczywistego	DS1307 RTC + CR2032	1	Z interfejsem I2C
3	Stabilizator napięcia	LM7805L	1	Zasilanie 5V z 12V
4	Wyświetlacz Nixie	IN-17	4	Cyfrowy, 10 cyfr
5	Sterownik Nixie	K155ID1	4	Dekoder BCD na 10 wyjść
6	Rezystor	10 kΩ	12	Ograniczanie prądu dla wyświetlaczy
7	Kondensator elektrolityczny	100 µF / 25V	2	Filtrowanie napięcia
8	Kondensator ceramiczny	100 nF	2	Odsprzęganie
9	Gniazda / złącza męskie i żeńskie	-	kilka	Do łatwego montażu/demontażu
10	Przewody połączeniowe	-	kilka	Do połączeń na płytce

Nr	Nazwa komponentu	Symbol / Model	Ilość	Uwagi
11	Płytka uniwersalna dwustronna	-	1	Do montażu wszystkich komponentów
12	Zasilacz	12V DC	1	Do zasilania całego układu
13	Przetwornica HV	170V DC boost converter	1	Do zasilania lamp Nixie

3.1 Lampy Nixie IN-17

Opis: Lampy Nixie IN-17 to klasyczne wyświetlacze wykorzystujące wyładowania jarzeniowe do prezentacji cyfr od 0 do 9. Każda cyfra posiada oddzielną katodę, która świeci po przyłożeniu wysokiego napięcia. Lampy te posiadają charakterystyczną szklaną obudowę cylindryczną.

3.2 Układy sterujące K155ID1

Opis: K155ID1 to układy scalone zaprojektowane do bezpośredniego sterowania lampami Nixie. Pozwalają na bezpieczne przełączanie wysokiego napięcia, wykorzystując niskonapięciowe sygnały logiczne z mikrokontrolera.

3.3 Stabilizator napięcia LM7805

Opis: Stabilizator liniowy 5 V, zasilany z wejściowego napięcia 12 V DC. Zapewnia stałe napięcie zasilania dla mikrokontrolera oraz pozostałych układów cyfrowych.

3.4 Arduino Pro Mini (ATmega328P)

Opis: Kompaktowa płytka mikrokontrolera oparta na układzie ATmega328P. Pełni rolę głównej jednostki sterującej – odpowiada za przetwarzanie sygnałów czasowych oraz sterowanie wyświetlaniem cyfr na lampach Nixie.

3.5 Moduł RTC DS1307

Opis: Zegar czasu rzeczywistego (RTC) z interfejsem I²C, umożliwiający dokładne śledzenie daty i godziny (sekundy, minuty, godziny, dzień tygodnia, data, miesiąc, rok). Moduł wyposażony jest w bateryjne podtrzymanie pracy..

3.6 Zasilacz wysokiego napięcia NCH6100HV

Opis: Moduł konwertujący niskie napięcie DC (np. 12 V) na napięcie rzędu 170 V DC, niezbędne do prawidłowego działania lamp Nixie. Zapewnia stabilne i bezpieczne zasilanie wyświetlaczy.

3.7 Uniwersalna płytka drukowana (PCB) 90 × 150 mm, dwustronna

Płytka prototypowa umożliwiająca montaż elementów poprzez lutowanie przewlekane oraz elastyczne rozmieszczenie połączeń elektrycznych na obu warstwach.

3.8 Przewody połączeniowe

Przewody sygnałowe i zasilające stosowane do łączenia poszczególnych modułów i komponentów na etapie montażu oraz prototypowania.

3.9 Złącza męskie i żeńskie (pin-header)

Standardowe złącza umożliwiające łatwe podłączanie i rozłączanie modułów. Znacznie ułatwiają testowanie, modyfikacje oraz serwisowanie układu.

3.10 Kondensatory SMD (rozmiar 1206)
Opis:
Elementy filtrujące stosowane w układzie zasilania – wygładzają napięcie oraz tłumią zakłócenia impulsowe.

3.11 Rezystory 10 kΩ

Stosowane m.in. jako rezystory ograniczające prąd w torach sterujących oraz jako elementy typu pull-up/pull-down.

4. Wycena Projektu

W poniższym rozdziale przedstawiono szczegółową wycenę komponentów wykorzystanych w projekcie zegara Nixie. Wszystkie elementy zostały zakupione na rynku polskim, co odzwierciedla rzeczywiste koszty realizacji projektu.

Wyjątkiem są lampy Nixie IN-17, które zostały pozyskane nieodpłatnie od znajomego z Ukrainy, pracującego w branży elektronicznej, gdzie często wyrzucane są użyteczne elementy. Dzięki temu udało się znacznie obniżyć koszty projektu.

Element	**Cena [zł]**
Płytka drukowana uniwersalna dwustronna 90×150 mm	9,90
Kondensator elektrolityczny 10 µF / 50 V, 5×11 mm, THT (2 szt.)	0,20
Kondensator ceramiczny 100 nF / 50 V, THT (4 szt.)	0,40
Zestaw przewodów 15 cm, czarne (100 szt.)	11,50
Konwerter USB-UART FTDI FT232RL miniUSB + przewód USB	39,90
Zestaw przewodów połączeniowych justPi – 20 cm, 3×40 szt. (m-m, ż-ż, m-ż)	17,50
Stabilizator 5 V L7805ABV – THT TO220	1,90
Arduino Pro Mini 328 – 3.3 V / 8 MHz – SparkFun DEV-11114	63,90
Moduł RTC DS1307 + 32 kb EEPROM 24C32 I2C	5,95
Wtyk goldpin 1×40 prosty, raster 2,54 mm, czarny (10 szt.) – justPi	3,90
Rezystor justPi THT CF węglowy 1/4 W, 10 kΩ (30 szt.)	1,90

Element	Cena [zł]
Zasilacz wysokiego napięcia 170 V DC NCH6100HV	41,59
Sterownik lamp Nixie K155ID1 (4 szt.)	80,00
Lampy Nixie IN-17 (4 szt.)	0,00*

| **Suma** | **279,34 zł**|

5. Kod

Poniżej przedstawiono pełny kod projektu zegara Nixie wraz z komentarzami wyjaśniającymi jego działanie:

```
#include <MD_DS1307.h>
#include <Wire.h>

// Definicje pinów sterujących lampami Nixie
#define A1 3
#define B1 4
#define C1 5
#define D1 6
#define A2 7
#define B2 8
#define C2 9
#define D2 10
#define A3 11
#define B3 12
#define C3 13
#define D3 14
#define A4 15
#define B4 16
#define C4 2
#define D4 1

#define pot A3 // Pin potencjometru (nieużywany w kodzie)

// Tablice pinów dla wygodnego sterowania
char A[4] = {A4, A3, A2, A1};
char B[4] = {B4, B3, B2, B1};
char C[4] = {C4, C3, C2, C1};
char D[4] = {D4, D3, D2, D1};

// Zmienne do przechowywania cyfr i czasu
int zero;
int one;
int two;
int three;
```

```
int hour;
int minute;

void setup() {

// Ustawienie pinów jako wyjścia
  pinMode(A1, OUTPUT);
  pinMode(B1, OUTPUT);
  pinMode(C1, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(A2, OUTPUT);
  pinMode(B2, OUTPUT);
  pinMode(C2, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(A3, OUTPUT);
  pinMode(B3, OUTPUT);
  pinMode(C3, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(A4, OUTPUT);
  pinMode(B4, OUTPUT);
  pinMode(C4, OUTPUT);
  pinMode(D4, OUTPUT);
  pinMode(pot, INPUT); // Potencjometr jako wejście

// Na początku wyłączamy cyfry na lampach (ustawiamy piny HIGH)
  for (char i = 0; i < 4; i++) {
    digitalWrite(A[i], HIGH);
    digitalWrite(B[i], HIGH);
    digitalWrite(C[i], HIGH);
    digitalWrite(D[i], HIGH);
  }

  // Inicjalizacja modułu RTC
  if (!RTC.isRunning())
    RTC.control(DS1307_CLOCK_HALT, DS1307_OFF);
Serial.begin(9600);
}

void loop() {
  RTC.readTime(); // Odczyt aktualnego czasu z RTC
  hour = RTC.h;
  minute = RTC.m;

  // Obliczenie cyfr do wyświetlenia
  zero = (hour / 10) % 10;
  one = hour % 10;
  two =  (minute / 10) % 10;
  three = minute % 10;
```

```
  // Debug - wypisanie wartości na port szeregowy
Serial.println(hour);
Serial.println(minute);
Serial.println(zero);
Serial.println(one);
Serial.println(two);
Serial.println(three);

offAll(); // Wyłączenie wszystkich cyfr na lampach

  // Wyświetlenie cyfr na poszczególnych lampach
  writenumber(0, zero);
  writenumber(1, one);
  writenumber(2, two);
  writenumber(3, three);

  delay(1000); // Odświeżanie co sekundę

}

// Funkcja wyświetlająca cyfrę b na lampie o indeksie a
void writenumber(int a, int b) {
  switch (b) {
    case 0:
      digitalWrite(A[a], LOW);
      digitalWrite(B[a], LOW);
      digitalWrite(C[a], LOW);
      digitalWrite(D[a], LOW);
      break;
    case 1:
      digitalWrite(A[a], HIGH);
      digitalWrite(B[a], LOW);
      digitalWrite(C[a], LOW);
      digitalWrite(D[a], LOW);
      break;
    case 2:
      digitalWrite(A[a], LOW);
      digitalWrite(B[a], HIGH);
      digitalWrite(C[a], LOW);
      digitalWrite(D[a], LOW);
      break;
    case 3:
      digitalWrite(A[a], HIGH);
      digitalWrite(B[a], HIGH);
      digitalWrite(C[a], LOW);
      digitalWrite(D[a], LOW);
      break;
```

```
    case 4:
      digitalWrite(A[a], LOW);
      digitalWrite(B[a], LOW);
      digitalWrite(C[a], HIGH);
      digitalWrite(D[a], LOW);
      break;
    case 5:
      digitalWrite(A[a], HIGH);
      digitalWrite(B[a], LOW);
      digitalWrite(C[a], HIGH);
      digitalWrite(D[a], LOW);
      break;
    case 6:
      digitalWrite(A[a], LOW);
      digitalWrite(B[a], HIGH);
      digitalWrite(C[a], HIGH);
      digitalWrite(D[a], LOW);
      break;
    case 7:
      digitalWrite(A[a], HIGH);
      digitalWrite(B[a], HIGH);
      digitalWrite(C[a], HIGH);
      digitalWrite(D[a], LOW);
      break;
    case 8:
      digitalWrite(A[a], LOW);
      digitalWrite(B[a], LOW);
      digitalWrite(C[a], LOW);
      digitalWrite(D[a], HIGH);
      break;
    case 9:
      digitalWrite(A[a], HIGH);
      digitalWrite(B[a], LOW);
      digitalWrite(C[a], LOW);
      digitalWrite(D[a], HIGH);
      break;
  }
}

// Wyłącza cyfrę na lampie o indeksie a (ustawia piny HIGH)
void off(int a) {
  digitalWrite(A[a], HIGH);
  digitalWrite(B[a], HIGH);
  digitalWrite(C[a], HIGH);
  digitalWrite(D[a], HIGH);
}
```

```
// Wyłącza wszystkie cyfry na wszystkich lampach
void offAll() {
  for (int i = 0; i < 4; i++) {
    digitalWrite(A[i], HIGH);
    digitalWrite(B[i], HIGH);
    digitalWrite(C[i], HIGH);
    digitalWrite(D[i], HIGH);
  }
}
```

Opis działania kodu

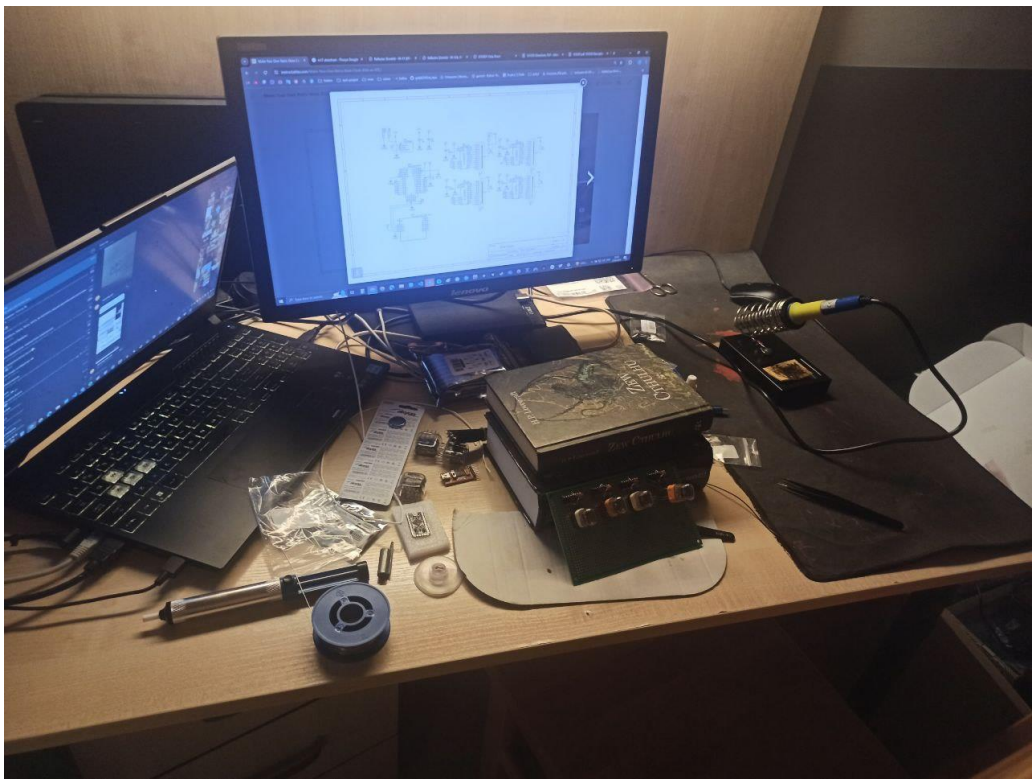
Przedstawiony kod odpowiada za sterowanie czterema lampami Nixie IN-17, wyświetlając aktualny czas pobierany z modułu RTC DS1307.

W funkcji setup() następuje inicjalizacja pinów oraz modułu zegara, a w pętli loop() co sekundę odczytywany jest aktualny czas i konwertowany na cyfry, które są przekazywane do funkcji writenumber(). Ta funkcja ustawia odpowiednie stany pinów, aby zapalić właściwą cyfrę na każdej lampie.

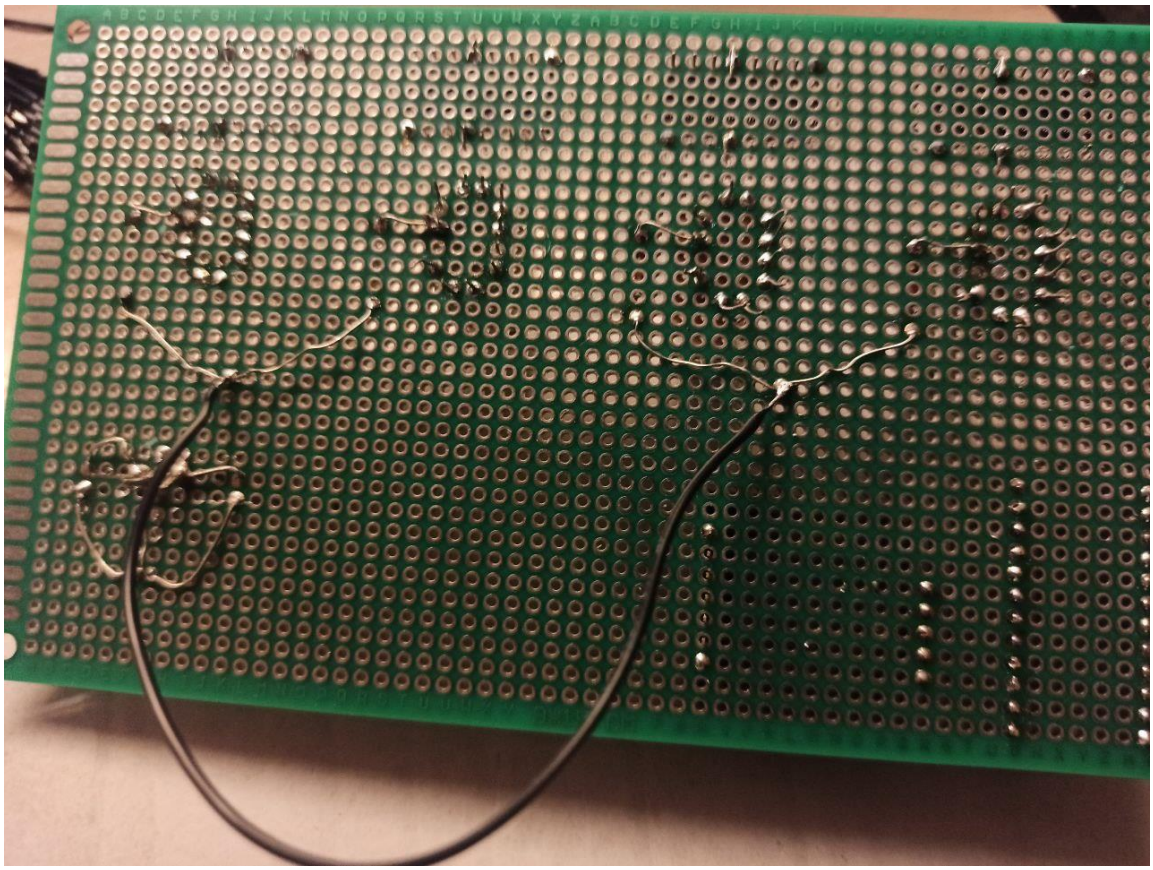
Dodatkowo, funkcje off() oraz offAll() służą do wyłączania poszczególnych cyfr lub wszystkich lamp, zapobiegając jednoczesnemu świeceniu wielu cyfr na jednej lampie.

6. Proces montażu i lutowania

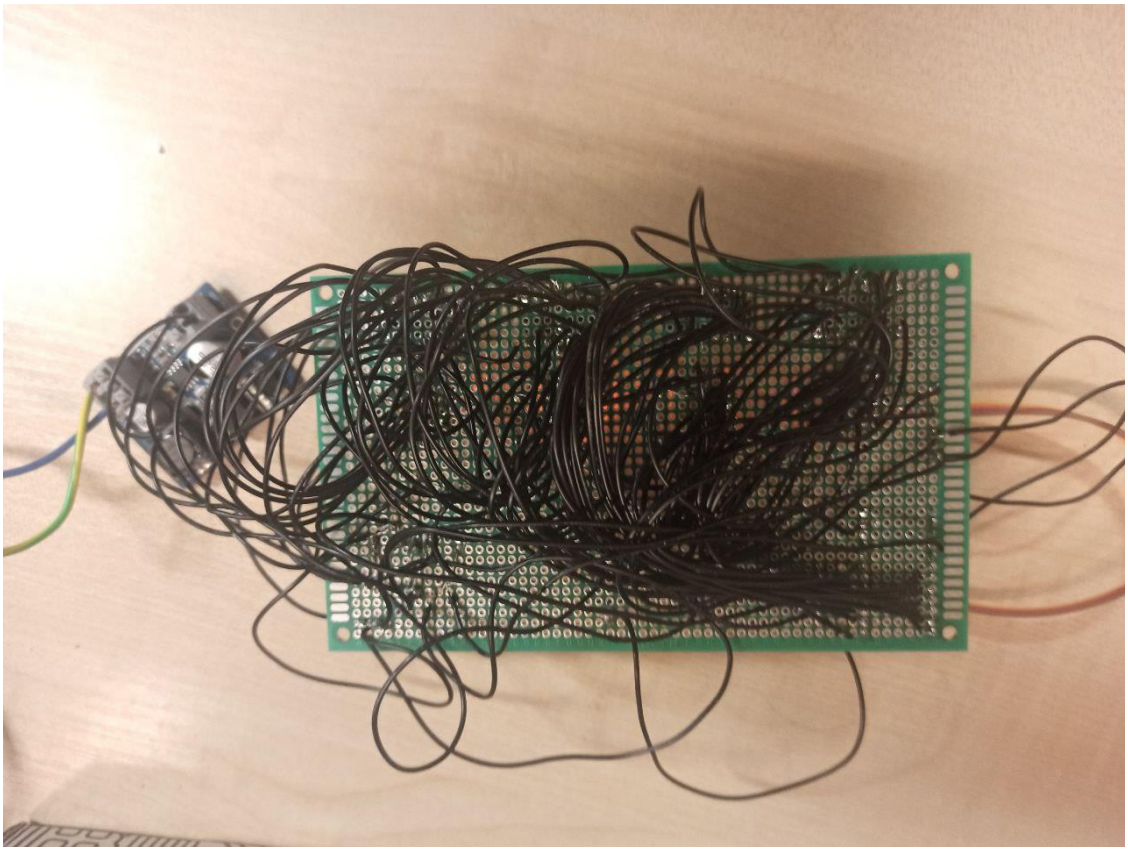
Poniższa seria zdjęć przedstawia kolejne etapy tworzenia zegara na lampach Nixie, wykonanego na dwustronnej płytce prototypowej



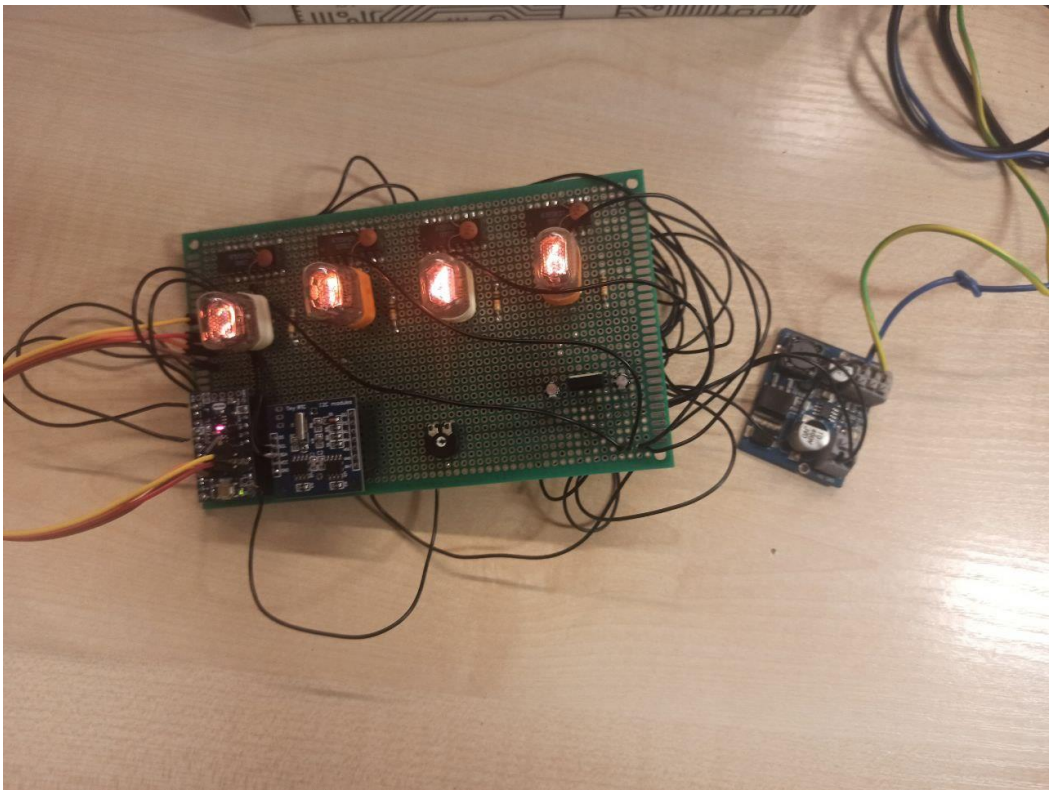
Na zdjęciu przedstawiono stanowisko pracy podczas montażu i lutowania zegara opartego na lampach Nixie.



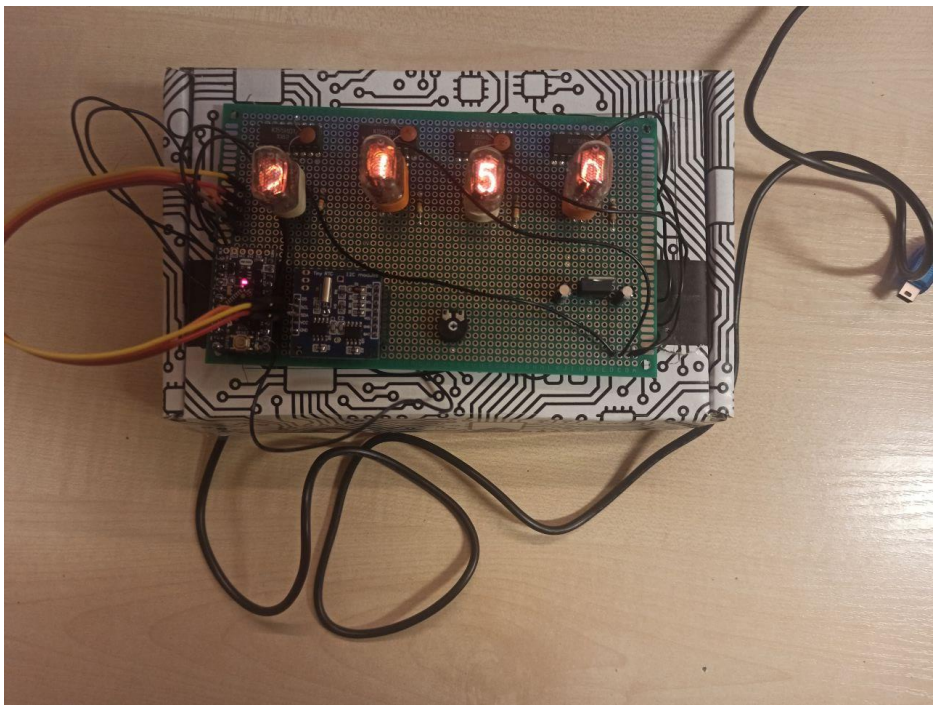
Zdjęcie przedstawia dolną stronę płytki uniwersalnej. Widać wczesny etap montażu.



Widok z góry na płytkę po zakończeniu większości połączeń.



Na zdjęciu widoczny jest uruchomiony układ zegara.



Zakończony etap montażu – płytka została umieszczona w tymczasowej obudowie.

Krótki film prezentujący test wyświetlania cyfr na lampach Nixie jest dostępny na YouTube: https://youtube.com/shorts/KTlzbwLbgn8?feature=share

6. Podsumowanie:

Głównym wyzwaniem podczas realizacji projektu było nieoptymalne rozplanowanie czasu pracy, co skutkowało koniecznością finalizowania go pod koniec semestru oraz rezygnacją z wykonania dedykowanej płytki drukowanej (PCB). Zamiast tego wykorzystano uniwersalną płytkę oraz przewody do połączeń, co wpłynęło na utrudnienia podczas precyzyjnego lutowania komponentów.

Po zakończeniu etapu montażu nie pojawiły się nowe problemy sprzętowe, a ewentualne trudności wystąpiły dopiero na etapie testowania oprogramowania sterującego Arduino oraz modułu RTC.

W trakcie testów zidentyfikowano kilka istotnych błędów:

- Druga lampa nie wyświetlała cyfr parzystych z powodu braku generowania niskiego sygnału na pinie 11 Arduino, co udało się rozwiązać wykorzystując potencjometr.
- Pierwsza lampa nie działała, ponieważ wykorzystywane były piny A6 i A7, które nie są przeznaczone do wyjścia sygnału. Po zmianie na odpowiednie piny lampy zaczęły funkcjonować prawidłowo.
- Problemy z konfiguracją RTC wynikały z błędnego ustawienia prędkości transmisji szeregowej (baud rate). Choć w kodzie ustawiono 9600, serial monitor wymagał ustawienia 4800. Ten problem został wyeliminowany metodą prób i błędów, co pozwoliło na poprawną konfigurację i stabilną pracę modułu RTC.

Po dołożeniu baterii podtrzymującej działanie RTC i przeprowadzeniu testów, zegar zachowuje poprawny czas nawet po wyłączeniu i ponownym uruchomieniu.

Wszystkie zdjęcia i filmy dokumentujące projekt — w tym proces montażu, etapy lutowania oraz końcowe testy — można znaleźć w odpowiednich folderach repozytorium GitHub: https://github.com/pavlokostushevych/Nixie-Tube-Clock.git