

# Arduino: Zegar cyfrowy z pomiarem temperatury

## Opis urządzenia

Projekt korzysta z Arduino oraz następujących elementów:

- **RTC\_DS1307** - moduł zegara czasu rzeczywistego (biblioteka RTCLib),
- **TM1637Display** - wyświetlacz 4-cyfrowy z podświetleniem - sterowany przez CLK i DIO,
- **DHT11** - sensor temperatury i wilgotności powietrza (podłączony do pinu 9),
- **DS18B20** - cyfrowy termometr (sensory podłączone do magistrali OneWire na pinie 3).

## Import bibliotek

- `RTCLib.h` - obsługa zegara RTC,
- `Arduino.h` - funkcje bazowe platformy Arduino,
- `TM1637Display.h` - sterowanie wyświetlaczem 4-cyfrowym,
- `DHT.h` - obsługa czujnika DHT11,
- `OneWire.h` i `DallasTemperature.h` - obsługa DS18B20,
- `SPI.h`, `Wire.h` - biblioteki komunikacyjne (`Wire™` używany przez RTC).

## Definicje pinów i obiekty

- `CLK 6`, `DIO 4` - piny wyświetlacza TM1637,
- `DHTPIN 9`, `DHTTYPE DHT11` - konfiguracja czujnika DHT11,
- `ONE_WIRE_BUS 3` - magistrala dla DS18B20.
- Obiekty:
  - `RTC_DS1307 rtc;`
  - `TM1637Display display(CLK, DIO);`
  - `DHT dht(DHTPIN, DHTTYPE);`
  - `OneWire oneWire(ONE_WIRE_BUS);`
  - `DallasTemperature sensors(&oneWire);`

## setup()

1. `rtc.begin()` - uruchamia zegar RTC,
2. `sensors.begin()` - inicjalizuje DS18B20,
3. Ustawia jasność wyświetlacza (`setBrightness`) i pokazuje „8888” jako test,
4. Inicjuje port szeregowy (`Serial.begin`) - do debugu pomiarów,
5. `dht.begin()` - inicjalizacja czujnika DHT11.

## loop() - cykl główny

1. `sensors.requestTemperatures()` i `getTempCByIndex(0)` - odczyt temperatury

DS18B20,

2. Konwersja na `int` - zaokrąglenie (`round`),
3. `rtc.now()` - pobiera aktualny czas; wyodrębnia godzinę i minutę,
4. Odczyt z DHT11: `readHumidity()` i `readTemperature()`,
5. Formatowanie czasu jako 4-cyfrowego stringa (np. „0835”),
6. Konwersja do liczby (`toInt()`) i wyświetlanie kolejno:
  - godzina i minuta,
  - temperatura z czujnika DHT11 (°C),
  - wilgotność z DHT11 (%),
  - temperatura z DS18B20 (wartość całkowita),

Każdy odczyt - 2 sekundy.

## Zastosowane moduły/biblioteki

- **RTClib** - obsługa modułów RTC DS1307/DS3231,
- **TM1637Display** - pokazanie cyfr na wyświetlaczu 4-segmentowym z kropką dziesiętną,
- **DHT** - odczyt wilgotności i temperatury z DHT11,
- **OneWire + DallasTemperature** - obsługa nawet wielu termometrów 1-wire,
- **SPI i Wire** - wymagane przez niektóre z powyższych (np. RTC).

## Przykład działania

1. Na starcie wyświetlacz pokazuje „8888” - test działania wszystkich segmentów,
2. Potem wyświetla czas (hhmm),
3. Następnie co 2 sekundy kolejno:
  - temperatura DHT11 (np. „23.5” → „23.5°C”),
  - wilgotność DHT11 (np. „45.2” → „45.2%”),
  - temperatura DS18B20 (np. „24” → 24°C).

## Możliwe modyfikacje

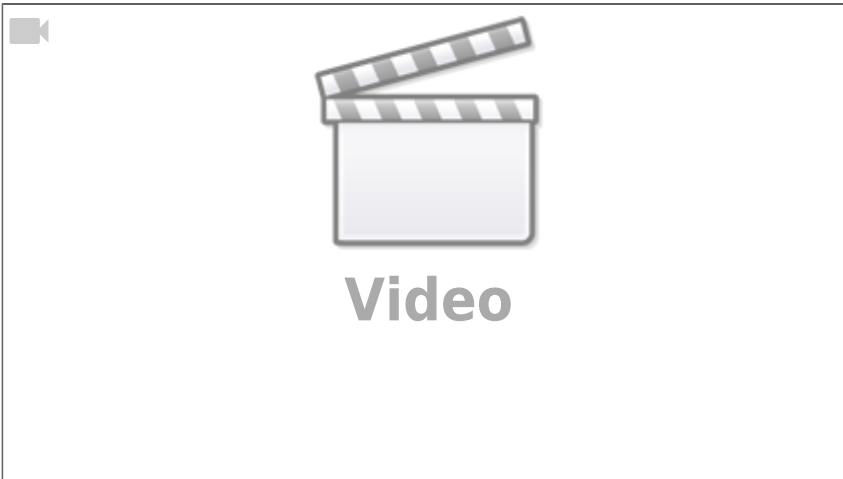
- Dodanie przycisków do ustawiania czasu,
- Rejestracja danych w EEPROM lub kartę SD,
- Rozbudowa interfejsu - np. menu wyboru wyświetlanych wartości,
- Zastosowanie DS3231 dla większej dokładności RTC.

## Podsumowanie

Urządzenie to zegar typu „cyfrowego” z pomiarem temperatury i wilgotności, wizualizujący wartości na wyświetlaczu TM1637. Korzysta z trzech różnych sensorów i prostych bibliotek Arduino, co czyni projekt świetnym przykładem integracji sprzętu i oprogramowania w jednym systemie.

## Filmik z prezentacją

Filmik nagrany dawno temu i po angielsku, więc proszę o wyrozumiałość ;)



## Kod

```
#include 'RTCLib.h'  
#include <Arduino.h>  
#include <TM1637Display.h>  
#include 'DHT.h'  
#include <OneWire.h>  
#include <DallasTemperature.h>  
#include <SPI.h>  
#include <Wire.h>  
  
#define CLK 6  
#define DIO 4  
#define DHTPIN 9  
#define DHTTYPE DHT11  
  
#define ONE_WIRE_BUS 3  
  
RTC_DS1307 rtc;  
  
TM1637Display display(CLK, DIO);  
  
DHT dht(DHTPIN, DHTTYPE);  
  
OneWire oneWire(ONE_WIRE_BUS);  
DallasTemperature sensors(&oneWire);  
  
void setup() {  
  
    rtc.begin();  
    sensors.begin();  
  
    display.setBrightness(0x0f);  
    display.showNumberDecEx(8888,0b01000000, false);  
    delay(3000);  
    display.clear();  
    display.setBrightness(3);  
}
```

```
Serial.begin(57600);
dht.begin();
}

void loop() {
  sensors.requestTemperatures();
  float tempC = sensors.getTempCByIndex(0);
  int tempINT = round(tempC);
  DateTime now = rtc.now();
  int Minute = now.minute();
  int Hour = now.hour();
  String strMinute;
  String strHour;
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if(Hour < 10)
    {strHour = '0' + String(Hour);}
  else if(Hour >= 10)
    {strHour = String(Hour);}
  else if(Hour == 0)
    {strHour = '00';}

  if(Minute < 10)
    {strMinute = '0' + String(Minute);}
  else if(Minute >= 10)
    {strMinute = String(Minute);}
  else if(Minute == 0)
    {strMinute = '00';}

  String Time;
  Time += strHour;
  Time += strMinute;

  if(Time == '00'){
    Time = '0000';
  }

  Serial.println(tempC);
  Serial.println(tempINT);
  int numTime = Time.toInt();
  display.clear();
  display.showNumberDecEx(numTime,0b01000000, true);
  delay(2000);
  display.clear();
  display.showNumberDec(t, false, 2, 1);
  delay(2000);
  display.clear();
  display.showNumberDec(h, false, 2, 1);
```

```
delay(2000);  
display.clear();  
display.showNumberDec(tempINT, false, 2, 1);  
delay(2000);  
  
}
```