

## Практичне заняття №7.

### Програмування годинника реального часу

**Мета:** дослідити можливості програмування годинника реального часу з використанням інтерфейсу I2C.

**Завдання:** зібрати схему і написати програму виведення на дисплей параметрів годинника, а саме: час, дата та ін.

**Обладнання:** плата Arduino; дисплей LCD 1602 (або 2004) з підключеним перехідником I2C; проводи; макетна плата; кабель USB; годинник реального часу DS3231.

### Загальні відомості

DS3231 – це годинник реального часу з екстремально точним ходом, завдяки вбудованому кварцовому резонатору з температурною компенсацією. Інтерфейс передачі даних – I2C. У мікросхемі є також вхід для підключення резервної батареї. При відключенні основного живлення мікросхема автоматично перемикається на роботу від резервної батареї, точність ходу від резервної батареї не порушується.

У DS3231 підтримується підрахунок секунд, хвилин, годин, днів місяця (дати), днів тижня, місяців і років (з урахуванням високосного року для місяців). Підтримується робота в 12 і 24 годинному форматі. Є 2 будильника можливістю їх налаштування і відстеження стану. Підстроювання точності температурної компенсації. А також два виходи – на 32 кГц (вихід становить 32.768 кГц) і програмований вихід від 1 Гц до 8.192 кГц. Є також вивід скидання – RST. Мікросхема випускається в корпусі SO-16.

Що важливо знати при роботі з мікросхемою годинника реального часу, так це ж як витягти з неї дані або записати їх туди. Годинник має інтерфейс I2C. Для того щоб здійснити запис даних (а це потрібно і для того, щоб прочитати дані), потрібно передати умову старту (ці команди здійснюються за рахунок апаратного або програмного I2C для мікроконтролера), далі передати адресу мікросхеми з бітом запису, далі

передати адресу регістру, до якого будемо звертатися і потім передати в цей регістр байт даних. Якщо слідом передати ще байт даних, він запишеться в наступний регістр і т.д. По закінченню потрібно передати умову зупинки.

Графічне зображення вищесказаного представлено на рис. 1:

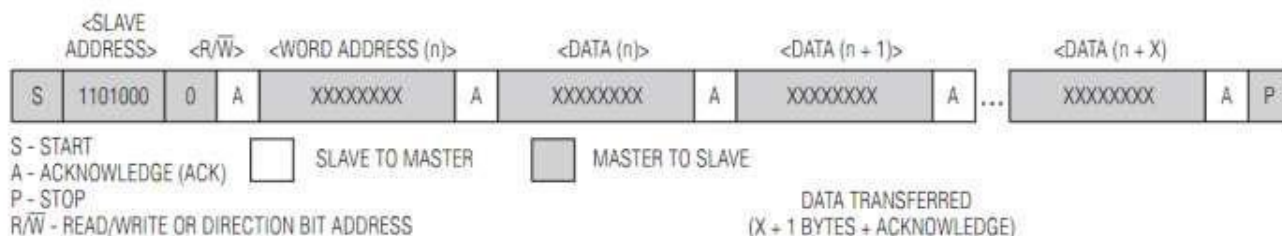


Рис. 1– Процес зчитування/запису для DS3231

Запис даних необхідний для початкового налаштування, а також для налаштування поточного часу. Далі нам потрібно постійно отримувати дані про поточний час і дату. Для цього необхідно здійснювати читання з регістрів зберігання цієї інформації. Читання складається з двох процедур – встановити покажчик на потрібний регістр і прочитати його. Щоб встановити покажчик на потрібний регістр, потрібно передати умову старту, потім передати адресу мікросхеми з бітом запису і байт з адресою регістру. Далі або умова зупинки і слідом умова старту, або просто рестарт. Тепер друга процедура – безпосередньо читання з регістрів. Старт переданий, далі потрібно відправити адресу мікросхеми з бітом читання і зчитувати регістри в необхідній кількості, по закінченню передати умову зупинки. Якщо інформація з регістра була прочитана, то покажчик автоматично переходить на наступний за ним регістр без зайвих дій з боку мікроконтролера (майстер пристрою). На рис. 2 проілюстровано все вище сказане щодо читання регістрів за рахунок I2C інтерфейсу.

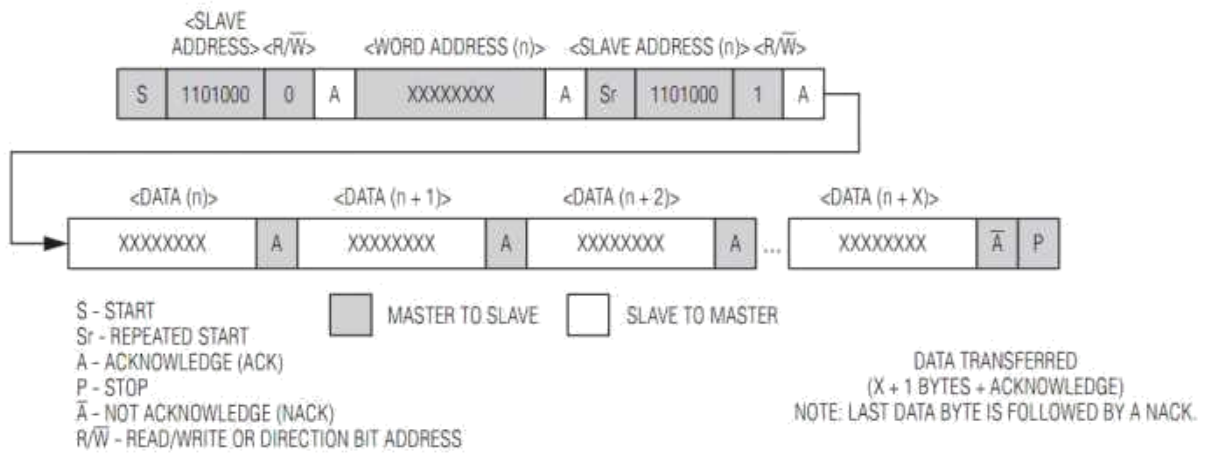


Рис. 2 – Читання регістрів за рахунок I2C-інтерфейсу

### Підключення годинника до Arduino

Годинник підключається по шині ІІС(I2C) (Inter-IntegratedCircuit) через виводи SDA і SCL до SDA і SCL Arduino відповідно. Також необхідно підключити живлення + 5В і GND.

Табл. 1 – Співвідношення виводів шини I2C в залежності від типу контролера, що розташовані на різних виводах

	Контакт SDA	Контакт SCL
На Arduino UNO, Nano, Pro Mini.	A4	A5
На Arduino Mega, Due.	20	21
На Arduino Leonardo, Pro Micro.	2	3

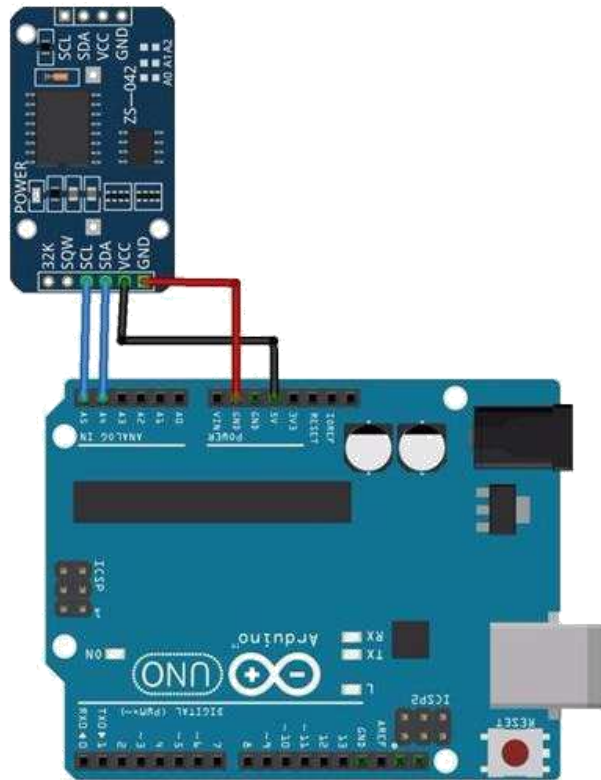


Рис. 3 – Схема підключення годинника реального часу DS3231

### Бібліотеки для роботи DS3231

Для підключення RTC годинника реального часу DS3231 було розроблено декілька бібліотек. Варто звернути увагу на `iarduino_RTC.h`. Вона універсальна (підходить для DS3231, DS1302, DS1307), а також використовую менше ОЗП та флеш-пам'яті.

#### Опис функцій і методів бібліотеки `iarduino_RTC.h`:

- `#include <iarduino_RTC.h>` – підключення бібліотеки.
- `iarduino_RTC` Об'єкт (НАЗВА [, ВИБІД\_RST [, ВИБІД\_CLK [, ВИБІД\_DAT]]]) – створення об'єкта.
- Функція `begin ()` – ініціалізація роботи RTC модуля.
- Функція `settime (СЕК [, ХВ [, ГОД [, ДЕНЬ [, МІС [, РІК [, ДТ]]]]])`; - встановлення часу.
- Функція `gettime (ПРЯДОК)` – отримання часу.

□ Функція *blinktime* (**ПАРАМЕТР [ЧАСТОТА]**) – змушує функцію *gettime* «блукати» зазначеним параметром часу.

Функція *period* (**ХВИЛИНИ**) – вказує мінімальний період звернення до модуля в хвилинах.

- *seconds* – повертає секунди від 0 до 59.
- *minutes* – повертає хвилини від 0 до 59.
- *hours* – повертає годинник від 1 до 12.
- *hours* – повертає годинник від 0 до 23.
- *midday* – повертає опівдні 0 або 1 (0-am, 1-pm).
- *day* – повертає день місяця від 1 до 31.
- *weekday* – повертає день тижня від 0 до 6 (0-неділю, 6-субота).
- *month* – повертає місяць від 1 до 12.
- *year* – повертає рік від 0 до 99.

#### Приклад програми:

У даному прикладі наведена програма виводу часу годинника DS1307 в Монітор порта.

```
#include <iarduino_RTC.h> // Підключаємо бібліотеку
iarduino_RTC time (RTC_DS1307); // Оголошуємо об'єкт time для
модуля на базі чіпа DS1307
void setup () {
    Serial.begin (9600); // Почнемо передачу даних в монітор
    послідовного порту
    time.begin (); // Ініціалізуємо RTC модуль
}
void loop () {
    if (millis () % 1000 == 0) { // якщо пройшла 1 секунда
        Serial.println (time.gettime ("d-m-Y, H: i: s, D")); //
        ВИВОДИМО час в монітор, одним рядком
        delay (2); // призупиняємо роботу скетчу на 2 мс.
    }
}
```

## **Хід виконання роботи**

1. Зібрати макет відповідно завдання.
2. Підключити схему до живлення(5В).
3. Завантажити програму в мікроконтролер Arduino.
4. Перевірити правильність роботи програми.

### **Завдання**

Зібрати схему зображену на рис.3. Виконати наступні завдання:

#### **1, 3 варіанти:**

Установити поточне значення часу на годинник та виводити його значення в Монітор послідовного порту кожні 5 сек.

#### **2, 5 варіанти:**

Установити поточне значення дати на годинник та виводити його в монітор порту за командою.

#### **4, 6 варіанти:**

Виставити значення часу та будильник. При спрацьовуванні останнього, виводити текстове повідомлення в Монітор послідовного порту.

**Підготувати звіт** згідно ДСТУ 3008-95 (лістинг програми, висновки, перелік посилань).

### **Контрольні питання**

1. Що таке DS3231, які має характеристики та який інтерфейс використовує?
2. Коротко опишіть принцип підключення годинника до Arduino.
3. Які бібліотеки існують для роботи DS3231?
4. Які основні функції має, описана в роботі, бібліотека DS3231?
5. Які значення та в яких діапазонах повертаються годинником?