

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Носуля Олега Павловича

(ПІБ)

академічної групи

122-19ск-2

(шифр)

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

освітньої програми

Комп'ютерні науки

(назва освітньої програми)

на тему:

*Розробка апаратного логгера для персональних
комп'ютерів на базі мікроконтролерів Arduino*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Приходченко С.Д.</i>			
розділів:				
спеціальний	<i>доц. Приходченко С.Д.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« »

2022 року

ЗАВДАННЯ
на кваліфікаційну роботу
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-19ск-2
(група)

Носуля Олега Павловича
(прізвище та ініціали)

тема кваліфікаційної роботи Розробка апаратного логгера для
персональних комп'ютерів на базі мікроконтролерів Arduino

затверджена наказом ректора НТУ «ДП» від « 18.05.2022 » № 268-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів проектно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	13.05.2022 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	27.05.2022 р.

Завдання видав

(підпис)

доц. Приходченко С.Д
(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Носуль О.П.
(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

РЕФЕРАТ

Пояснювальна записка: 60 с., 23 рис., 4 таблиці, 5 дод., 20 джерел.

Об'єкт розробки: система ідентифікування працівників через RFID мітки.

Мета кваліфікаційної роботи: розробка та адміністрування кінцевого приладу.

У вступі розглядається аналіз та стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточняється постановка завдання.

У першому розділі проведено аналіз предметної галузі, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до реалізації, технологій та засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми та приладу, наведено опис алгоритму і структури функціонування системи, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробки приладу, проведений підрахунок вартості роботи по створенню приладу та розраховано час на його створення.

Практичне значення полягає у створенні системи, що забезпечує працівникам полегшений та захищений доступ до робочих ПК .

Актуальність приладу забезпечує широке розповсюдження RFID міток, що використовуються для пропуску та фіксації працівників і не потребує запроваджувати нові мітки використовуючи вже існуючі.

Список ключових слів: RFID, ДОСТУП , ПРИЛАД , ПК.

ABSTRACT

Explanatory note: 60 pp., 23 figs., 4 lists, 5 apps, 20 sources.

Object of development: system of identification of employees through RFID tags.

The purpose of thesis: development and administration of the final device.

In the introduction to the analysis and state of the problem, clarifying the purpose of the qualification work and areas of its application, provides a justification of current topics and clarifies the task.

In the first draw the analysis of the subject branch is carried out, the actual application and recognition of development is shown, the statement of the task is developed, opportunities for realization, technology and needs are set.

In addition, analyzes of various systems were conducted, platforms for development, project development and development of programs and applications were selected, a description of algorithm and structural-functional system was introduced, external and initial data were determined, warehouse parameters of technical requirements were specified.

The economic section indicates the complexity of raw material development, calculates the cost of work on the creation of raw materials and calculates the time for its creation.

The practical importance of the field in creating a system that provides benefits to employees and protects access to work PCs.

The relevance of the application is protected by the widespread use of RFID tags, which are maintained to pass and capture employees and do not consume the introduction of new tags using the same.

Keywords: RFID, ACCESS, DEVICE, PC.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ....	9
1.1. Загальні відомості з предметної галузі.....	9
1.2. Призначення розробки та галузь застосування	9
1.3. Підстави для розробки	9
1.4. Постановка завдання	10
1.5. Вимоги до програми або програмного виробу	10
1.5.1. Вимоги до функціональних характеристик	10
1.5.2. Вимоги до інформаційної безпеки.....	10
1.5.4. Вимоги до інформаційної та програмної сумісності	11
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ .	12
2.1. Функціональне призначення інформаційної системи	12
2.2. Опис застосованих математичних методів	12
2.3. Опис використаних технологій та програмних продуктів	12
2.3.1. Arduino	12
2.3.2. Arduino Leonardo.....	14
2.3.2. Real Time Clock модуль DS1307	18
2.3.1. Модуль microSD карти на базі LVC125A	19
2.3.2. Модуль RFID на базі RC522	22
2.3.3. Бібліотеки Mouse та Keyboard.....	24
2.3.4. Бібліотека SPI.....	25
2.3.5. Бібліотека Wire	25
2.3.6. Бібліотека TimeLib	26
2.3.7. Бібліотека DS1307RTC.....	26
2.3.8. Бібліотека MFRC522	26
2.4. Опис структури системи та алгоритмів її функціонування	27
2.5. Обґрунтування та організація вхідних та вихідних даних	28
2.6. Опис розробленої інформаційної системи.....	28
2.6.1. Використані технічні засоби	28

2.6.2. Використані програмні засоби засоби.....	28
2.6.2.1. Arduino IDE	28
2.6.2.2. Fritzing.....	35
2.6.3. Налаштування та підключення приладу	Ошибка! Закладка не определена.
2.6.4. Опис інтерфейсу користувача	41
РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ	42
3.1. Розрахунок трудомісткості та вартості розробки інформаційної системи....	42
3.2. Розрахунок витрат на створення інформаційної системи	46
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТОК А КОД ПРОГРАМИ	52
ДОДАТОК Б СХЕМА РОЗРОБЛЕНОГО ПРИЛАДУ	57
ДОДАТОК В ЗОВНІШНІЙ ВИГЛЯД РОЗРОБЛЕНОГО ПРИЛАДУ	58
ДОДАТОК Г ВІДГУК КЕРІНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	59
ДОДАТОК Д ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- RFID - Radio Frequency IDentification;
- ПК - персональний комп'ютер;
- RTC - Real-time clock;
- SPI - Serial Peripheral Interface;
- МК - Мікроконтроллер.

ВСТУП

На поточний день все більше компаній що мають офісні відділи та не тільки інтегрують RFID картки для контролю та реєстрації робітників за часом коли ті з'являються на роботу та закінчують її, що дозволяє автоматизувати звітність і не тільки. Проте це не єдиний напрямок де можна використати такі картки, що й нашоєхує на розробку цієї кваліфікаційної роботи [19].

Галуззю застосування об'єкта розробки є переважно офіси компаній з великою кількістю ПК.

Тематика даної кваліфікаційної роботи присвячена розробці пристрою на базі Arduino для полегшеного захищеного доступу до робочих ПК.

Основною ідеєю цієї кваліфікаційної роботи є використання RFID міток робітників компанії для доступу до ПК. Індивідуальний підхід до використання міток дозволяє реалізувати певним працівникам доступ лише до певних ПК, в той же час забороняючи доступ тим чи іншим робітникам.

Метою даної кваліфікаційної роботи є створення зручної інформаційної системи, що дозволить реалізувати захищений доступ до персональних комп'ютерів.

Для реалізації поставленої мети необхідно виконати наступні завдання:

- проаналізувати предметну галузь;
- визначити основні вимоги до інформаційної системи;
- обрати необхідний мікроконтролер;
- проаналізувати варіанти та вирішити які інструменти використовувати для розробки;
- розробити кінцевий прилад.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

Все більше і більше компаній обирають RFID мітки як засіб для моніторингу відвідуваності роботи своїми працівниками. Проте це не єдиний напрямок де можна використати такі картки, що й наштовхує на розробку цієї кваліфікаційної роботи. У цей же час набирає оборотів процес корпоративного шпіонажу та крадіжки важливої робочої інформації як самими працівниками, так і через їх необачливість та шляхом соціальної інженерії.

1.2. Призначення розробки та галузь застосування

Під час виконання кваліфікаційної роботи було поставлено завдання розробити інформаційну систему «Апаратний логер для персональних комп'ютерів на базі мікроконтролерів Arduino».

Галузь застосування даного продукту – будь які офіси чи інші структури з обширною кількістю ПК [19].

Призначення даної розробки - це реалізація системи з надавання доступу до персональних комп'ютерів працівників лише за допомогою їх індивідуальних міток, що і без використання цього приладу вже давно інтегровані в більшість компаній.

1.3. Підстави для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 268-с від 18.05.2022 р;
- завдання на кваліфікаційну роботу на тему «Розробка апаратного логгера для персональних комп'ютерів на базі мікроконтролерів Arduino».

1.4. Постановка завдання

Метою кваліфікаційної роботи є створення зручного пристрою для контролю та полегшення доступу до робочих ПК.

Призначення інформаційної системи - забезпечити зручний доступ , що надає можливість робітникам використовувати свої RFID мітки не лише для реєстрації на вході.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Кінцева інформаційна система повинна дотримуватися наступних функціональних вимог:

- можливість легкого підключення;
- ідентифікатори повинні перевірятися коректно;
- за необхідності записувати данні щодо наданого доступу до внутрішньої пам'яті пристрою.

1.5.2. Вимоги до інформаційної безпеки

Для усунення некоректної роботи інформаційної системи необхідно реалізувати:

- реєстрація користувачів повинна бути схована від робітників та доступна лише адміністраторам;

- семантичний та синтаксичний контроль вхідних даних;
- обробку виняткових ситуацій;
- можливість безперервної роботи протягом не менше 120 годин (5 діб);
- платформну незалежність.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для підтримки роботи приладу, слід дотримуватися таким технічним вимогам:

- наявність відключення до ПК через USB шину;
- налаштованість приладу системним адміністратором;
- наявність відповідних RFID міток;
- наявність SD карти для зберігання звітності приладу.

1.5.4. Вимоги до інформаційної та програмної сумісності

Інформаційна система потребує від користувача мати середовище з такими складовими: одна з операційних систем Windows, Linux, Mac OS.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення інформаційної системи

Розроблена інформаційна система призначена для перевірки ідентифікаторів RFID міток декількох користувачів (працівника або групи працівників та адміністратора). Має такі функціональні можливості:

- Надання доступу до системи;
- Запис звітності щодо взаємодій з приладом;
- Ігнорування усіх інших міток.

2.2.Опис застосованих математичних методів

У даному програмному продукті нема потреби використовувати математичні методи, оскільки більша частина роботи полягає у перевірці на збіги з заздалегідь заданими ідентифікаторами.

2.3.Опис використаних технологій та програмних продуктів

2.3.1. Arduino

Як відомо, Ардуіно придумали в Італії, оригінальні плати там і роблять.

Самі італійці випускають плату в декількох основних форм-факторах:

- Arduino - стандартний розмір, 20 вхідно-виходів, повна сумісність з усіма ШІлд (рис 2.1.).

- ArdinoMega - збільшений розмір, 70 вхідно-виходів, сумісність не з усіма ШІлд.

- ArdinoNano - Зменшення розмірів, 22 вхідно-виходів не сумісна з ШІлд.

- ArdinoMini - ще менший розмір, 20 вхідно-виходів, не сумісна з ШІлд, не має USB.

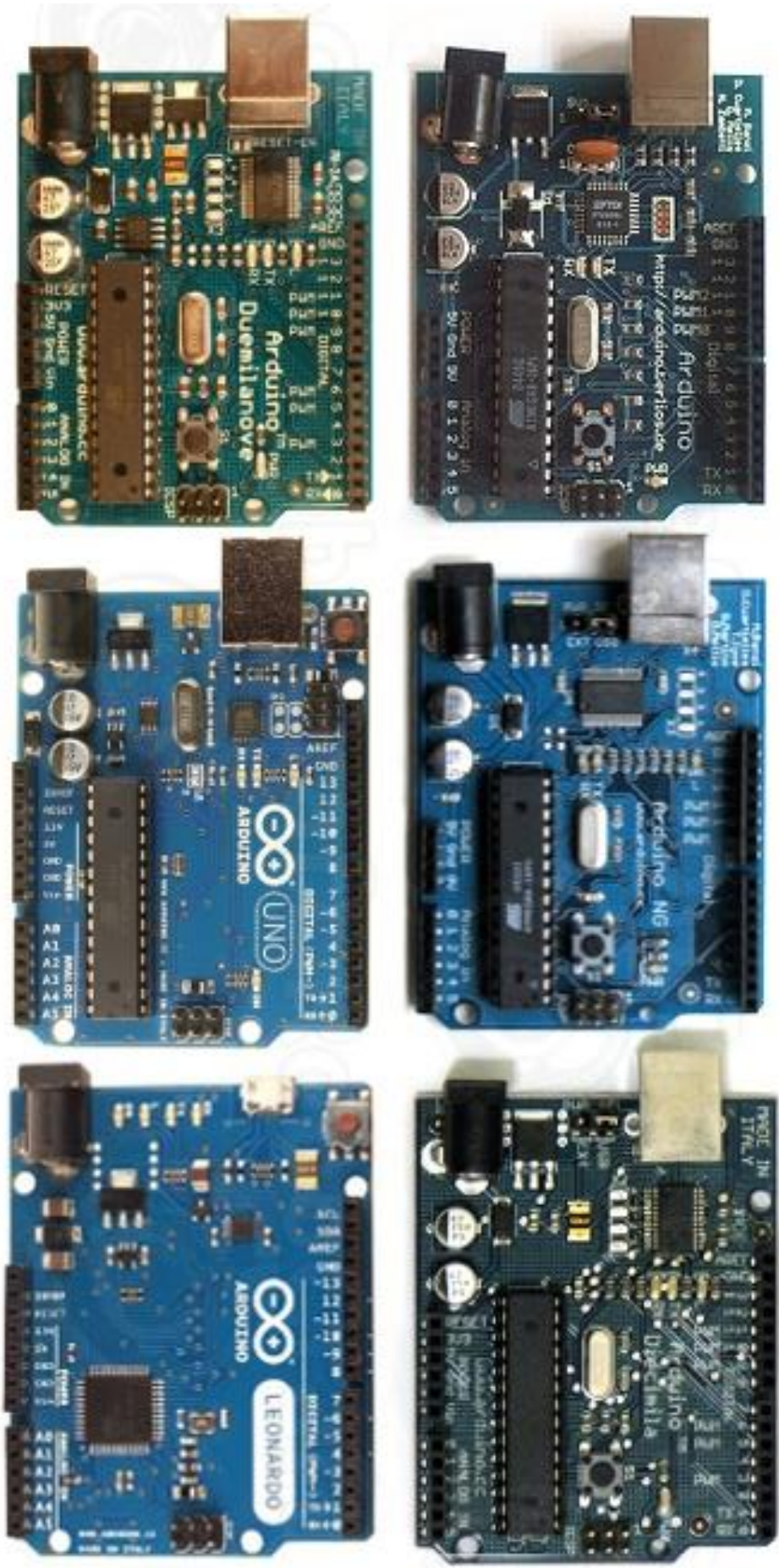


Рис 2.1. Форм-фактор Arduino xxx

Стандартний і найпоширеніший розмір. Коли говорять «Ардуіно» («звичайна Ардуіно») - зазвичай все відразу уявляють саме такі плати (рис. 2.1.).

Найперші плати були в цьому цьому форм-факторі, відповідно саме він пережив найбільше реінкорнацій (USB-версії в хронологічному порядку виходу): Extreme, NG, Diecimila, Duemilanove, Uno, Leonardo.

Всі ці плати мають однакову кількість входів-виходів, зібраних на однакові роз'єми (для підключення периферії і ШІД), програмуються по USB, і мають мікроконтролер АТМega на борту. На ранніх версіях стояв АТМega8, потім стали ставити АТМega168, потім АТМega328.

На «вісімці» тільки 3 ШІМ виходів, 8Кб під скетч 1Кб оперативної пам'яті, але для багатьох додатків вистачає. У АТМega168 вже 6 ШІМ каналів і 16Кб під ваші потреби, а у 328-й 32Кб під програми і вже 2Кб оперативної пам'яті. До речі не вся флеш-пам'ять доступна користувачу, частина її займає бутлоадер.

На всіх платах до UNO стояв чіп-перетворювач USB-UART FT232, що дозволяє встромляти плату прямо в USB і програмувати без програматора. При підключенні в системі з'являвся віртуальний COM-порт, який і використовується середовищем розробки Ардуіно для програмування [2].

2.3.2. Arduino Leonardo

Плата побудована на АТmega32u4 і в порівнянні з попередніми моделями покращилася. На 0,5кБ збільшилася оперативна пам'ять, ШІМ-виходів стало на 1 більше, аналогових входів стало 12 (6 розташовані де у всіх Ардуіно-плат, нові +6 розкидані по цифровим пінам) і розділені USB і UART. Так само невігадливо підтримуються, не тільки віртуальний ком порт, але і миша і клавіатура, набагато простіше ніж це реалізовано в UNO [6][18].

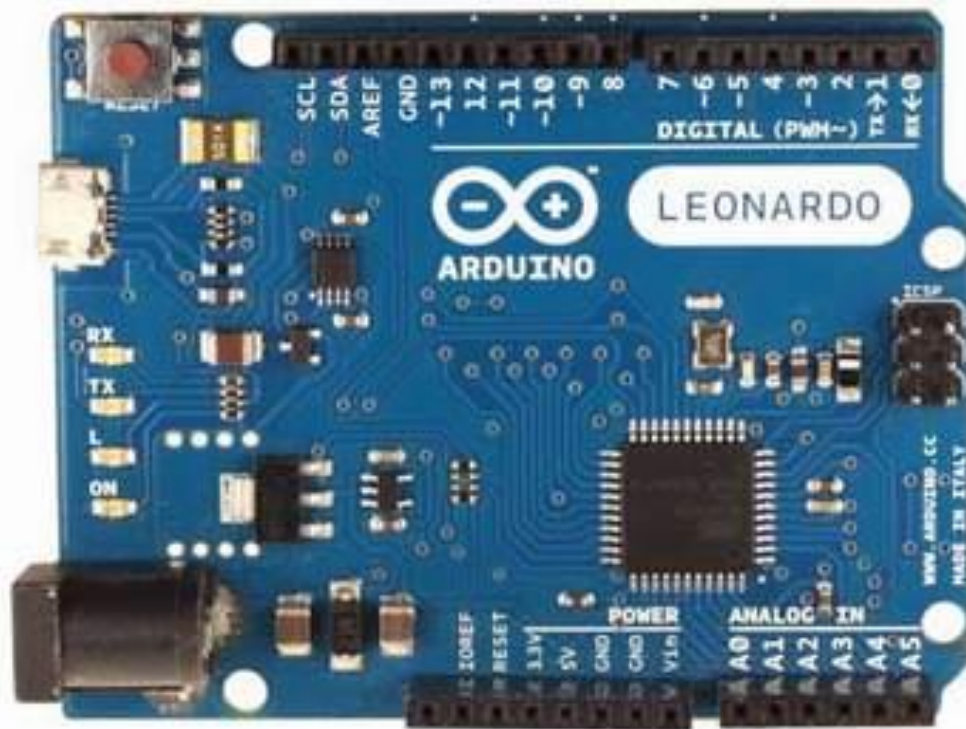


Рис. 2.2. Arduino Leonardo

Характеристики Arduino Leonardo наведені у таблиці 2.1. [14].

Таблиця 2.1

Загальні характеристики Arduino Leonardo

МК	ATmega32u4
Розрядність	8 біт
Тактова частота	16 МГц
Напруга логічних рівнів	5 V
Вхідна напруга живлення	7-12 V
Загальних портів вводу-виводу	20
Максимальний I пінів вводу-виводу	40 mA
Максимальний I піна 3.3V	50 mA
Максимальний I піна 5V	800 mA
ШІМ порти	7
АЦП порти	12
Розрядність АЦП	10 біт
Flash-пам'ять	32 KB

EEPROM-пам'ять	1 KB
ОЗУ	2,5 KB
Габарити	69x53 мм

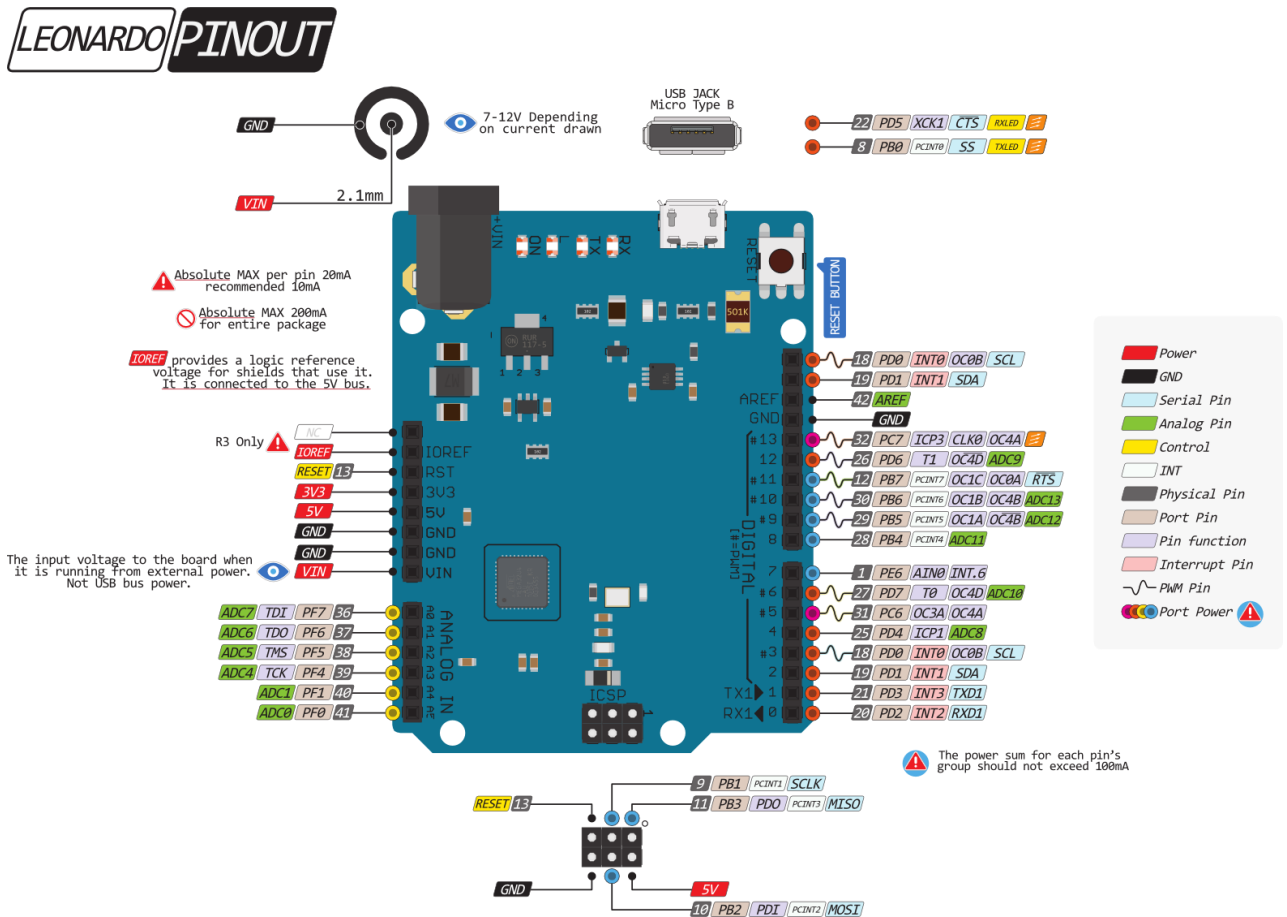


Рис. 2.3. Схема пінів для Arduino Leonardo

Згідно до рис. 2.3. ця плата має на своєму борту наступні піни:

- VIN: Напруга від зовнішнього джерела живлення (не пов'язана з 5 В від USB або іншою стабілізованою напругою). Через цей вихід можна подавати зовнішнє живлення, так і споживати струм, коли пристрій запитано від зовнішнього адаптера.

- 5V: На вихід надходить напруга 5 від стабілізатора плати. Стабілізатор забезпечує живлення мікроконтролера ATmega32U4. Живити пристрій через виведення 5V не рекомендується - у цьому випадку не використовується стабілізатор напруги, що може призвести до виходу плати з ладу.

- 3.3V: 3,3 В від стабілізатора напруги плати. Максимальний струм – 50 мА.

- GND: Виходи землі.

- IOREF: Цей вихід надає платам розширення інформацію про робочу напругу мікроконтролера. Залежно від напруги, плата розширення може переключитися на відповідне джерело живлення або задіяти перетворювачі рівнів, що дозволить їй працювати як з 5, так і з 3,3 пристроїв.

- Цифрові входи/виходи: піни 0-13 логічний рівень одиниці – 5 В, нуля – 0 В. Максимальний струм виходу – 40 мА. До контактів підключені резистори, що підтягують, які за замовчуванням вимкнені, але можуть бути включені програмно.

- ШИМ: піни 3,5,6,9,10,11 та 13 дозволяють виводити 8-бітні аналогові значення у вигляді ШИМ-сигналу.

- АЦП: піни А0-А5, А6-А11 (на цифрових пінах 4, 6, 8, 9, 10 та 12). Arduino Leonardo має 12 аналогових входів, кожен з яких може уявити напругу у вигляді 10-бітного коду (1024 значень). Розрядність АЦП - 10 біт.

- TWI/I²C: піни SDA та SCL для спілкування з периферією за синхронним протоколом, через 2 дроти з використанням бібліотеки Wire.

- SPI: піни роз'єму ICSP дозволяють здійснювати зв'язок за інтерфейсом SPI. Зверніть увагу, що лінії SPI виведені лише на роз'єм ICSP і не з'єднані з виходами плати, як Arduino Uno. Ті SPI-плати розширення, які не мають 6-контактного роз'єму ICSP для приєднання до Leonardo - працювати не будуть.

- UART: піни 0(RX) та 1(TX) використовуються для комутації плати Arduino із іншими пристроями через клас Serial1. Для зв'язку Arduino Leonardo із комп'ютером через порт micro-USB, використовуйте клас Serial.

- ICSP-роз'єм. Призначений для внутрішньосхемного програмування мікроконтролера ATmega32U4. Також із застосуванням бібліотеки SPI дані виходи можуть здійснювати зв'язок з інтерфейсом SPI. Зверніть увагу лінії SPI не продубльовані на цифрових контактах, наприклад у Arduino Uno [4].

2.3.2. Real Time Clock модуль DS1307

Модуль DS1307 з годинником реального часу. Модуль поставляється повністю зібраним та з запрограмованим поточним часом (встановіть свій часовий пояс перед використанням в проєкті). За допомогою літієвої батареї (CR2032-210mAh) модуль може працювати не менше 5 років без додаткового джерела живлення 5В [9].

Крім мікросхеми годинника реального часу модуль містить мікросхему I2C EEPROM 24C32 та інтерфейс для підключення датчика температури DS18B20. Чіп енергонезалежної пам'яті EEPROM дозволяє зберігати дані отримані від сенсора локально на модулі без необхідності постійно «смикати» мікроконтролер. Характеристики модулю наведені у таблиці 2.2. [1][11][17].

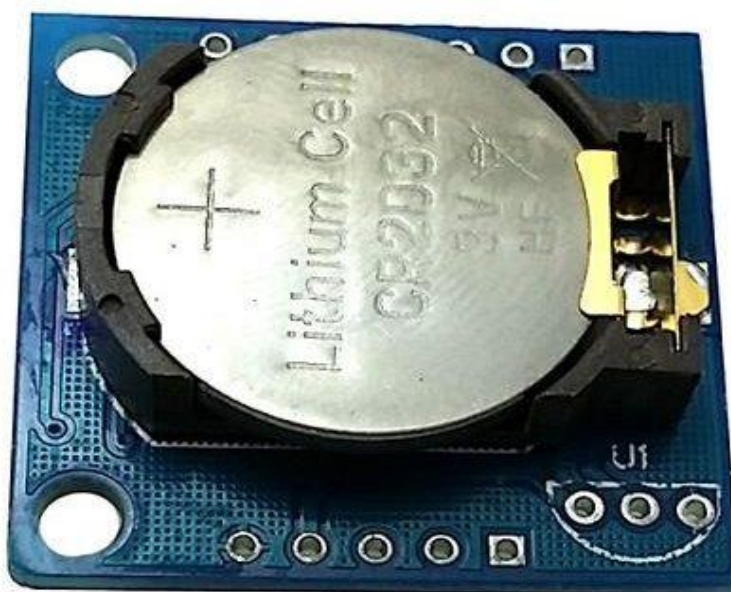


Рис.2.4. Зовнішній вигляд модулю реального часу RTC DS1307

Характеристики RTC модулю на базі DS1307

Інтерфейс	I2C
Формат часу	Години : Секунди AM/PM
Формат дати	Day Month, Date – Year
Врахування високосних років	Присутнє
Прораховані дати	До 2100 року
Форм-фактор батареї	CR2032
Енергонезалежна пам'ять	56b

2.3.1. Модуль microSD карти на базі LVC125A

Модуль підключення micro-SD карт для використання карт в якості файлового зберігання інформації. На платі розміщені стабілізатор 3.3В та перетворювач логічних рівнів 74LVC125, що робить модуль дуже універсальним і дозволяє підключати його як до 5В контролерів Arduino UNO, так і до 3.3В Arduino DUE. Характеристики модулю наведені у таблиці 2.3. [8] [1][16].

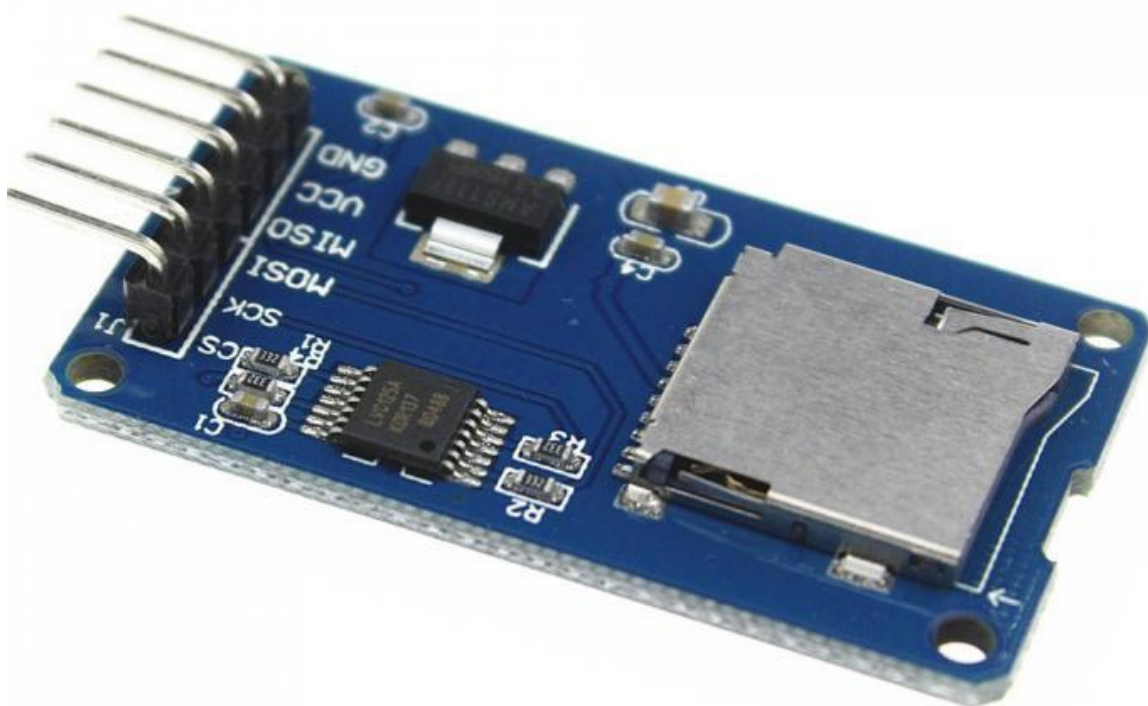


Рис 2.5. Зовнішній вигляд модулю microSD карти

Таблиця 2.3

Характеристики модулю microSD карти на базі LVC125A

Інтерфейс	SPI
Напруга живлення	4.5-5.5 V
Струм	0.2-200 mA
Логічні рівні	3,3V 5 V
Карти що підтримуються	Micro SD <=2 Gb Micro SDHC <=32 Gb
Розмір	42x24x12 mm

У ході кваліфікаційної роботи до схеми модулю були внесені незначні зміни через відносно погану якість виробництва. Всі переробки показані на рис. 2.6-2.8 де фіолетовим кольором зазначена доріжка плати, що була перебита, а зеленим – місця де був доданий кабель-перемичка.

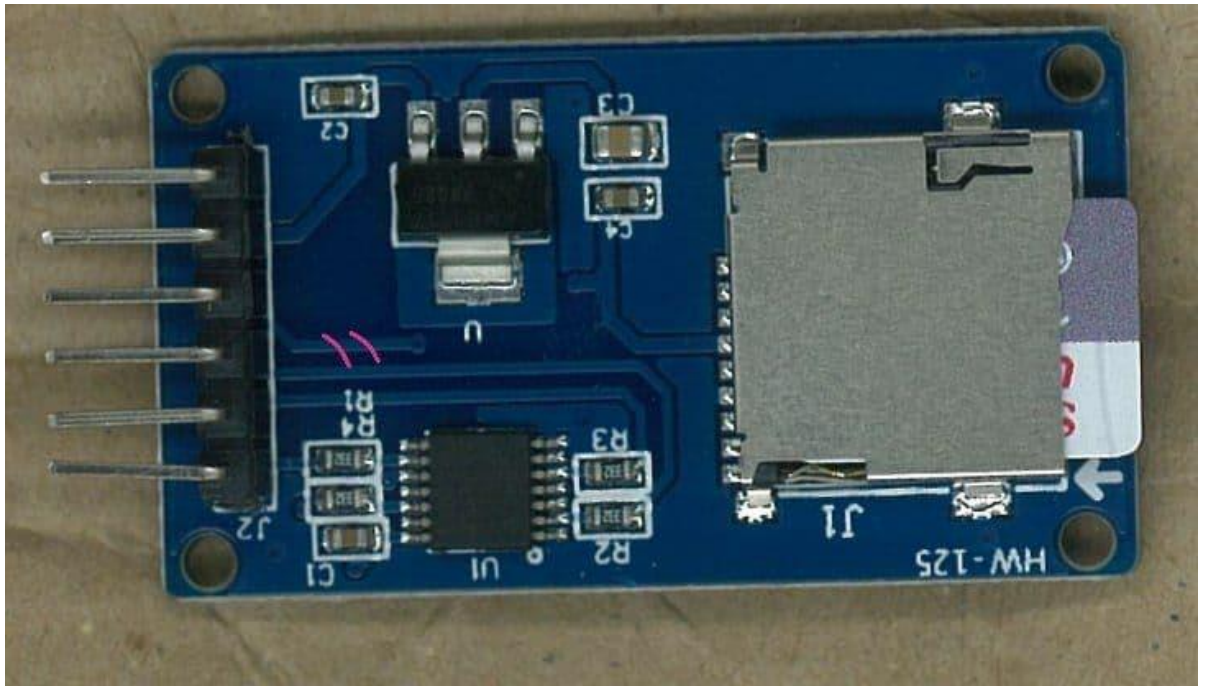


Рис. 2.6. Схема переробки модулю microSD карти (перебита доріжка)

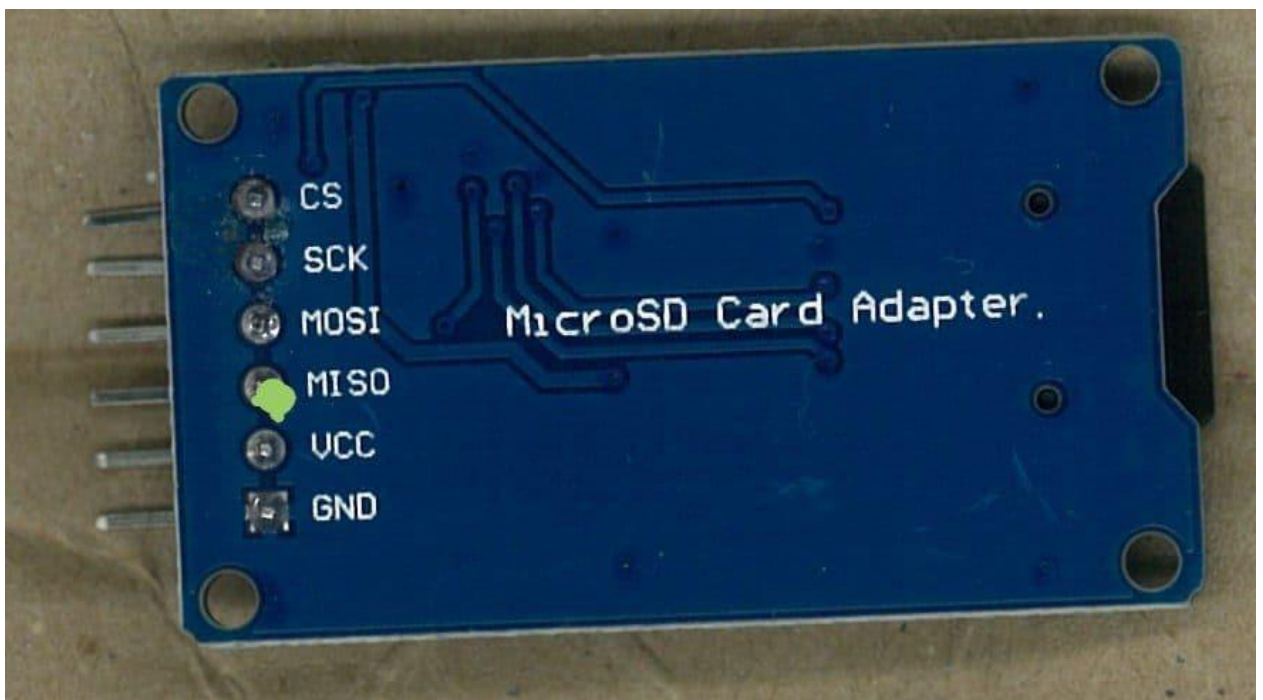


Рис. 2.7. Схема переробки модулю microSD карти (перша точка з'єднання кабелю-перемички)

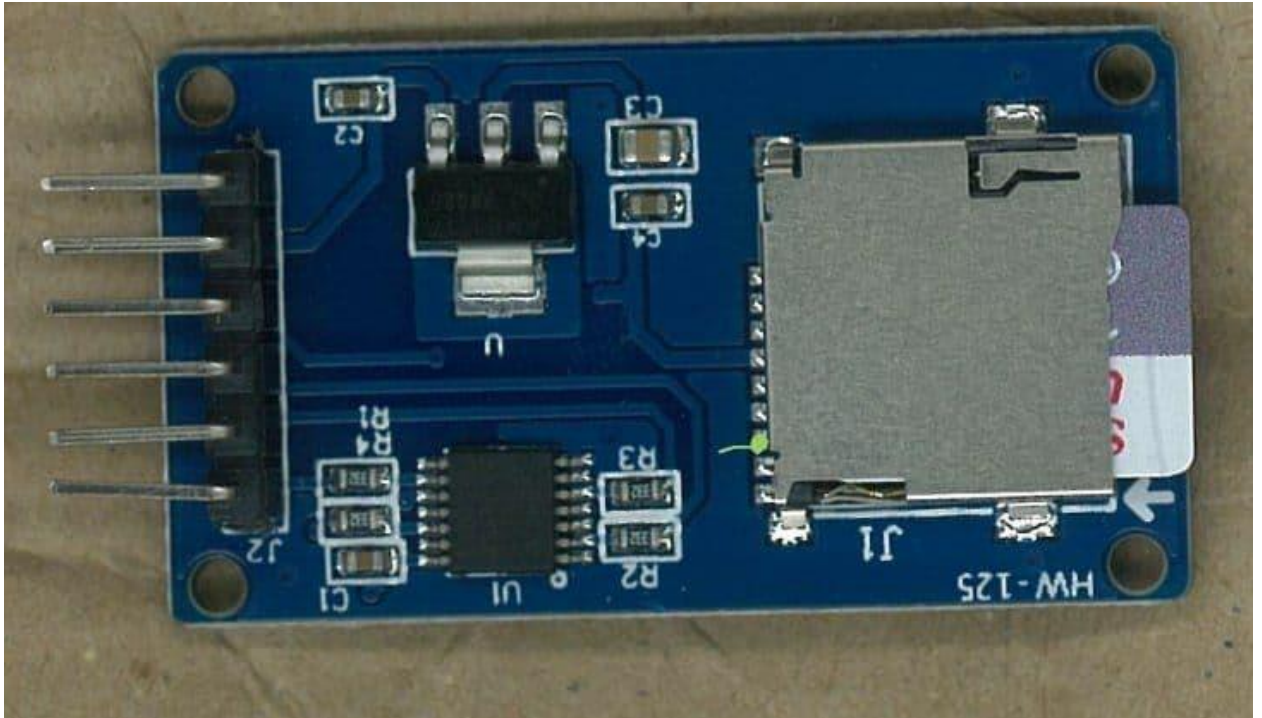


Рис 2.8. Схема переробки модулю microSD карти (друга точка з'єднання кабелю-перемички)

2.3.2. Модуль RFID на базі RC522

RFID-модуль 13.56 МГц з SPI-інтерфейсом. Даний модуль може бути використаний для різних радіоаматорських і комерційних застосувань, в тому числі для контролю доступу, автоматичної ідентифікації, робототехніки, відстеження речей, платіжних систем і т.д. Характеристики модулю наведені у таблиці 2.4. [7][12][15].



Рис 2.9 Зовнішній вигляд RFID модулю

Таблиця 2.4

Характеристики RFID модулю RC522

Інтерфейс	SPI
Мікросхема	MFRC522
Напруга живлення	3.3 V
Струм	13-26 mA
Струм у режимі очікування	10-13 mA
Струм у режимі сну	<80 mA
Робоча частота	13.56 MHz
Дальність реакції	0-16 мм
Розміри	40x60 мм

2.3.3. Бібліотеки Mouse та Keyboard

Ці базові бібліотеки дозволяють платам Arduino Leonardo, Micro або Due при підключенні до комп'ютера визначатися як звичайна миша та/або клавіатура.

Mouse має функції для роботи з мишею, що дозволяють Leonardo, Micro або Due контролювати рух курсору на підключеному комп'ютері. Оновлення позиції курсору завжди здійснюється щодо його попереднього положення. До бібліотеки Mouse входять наступні функції:

- Mouse.begin();
- Mouse.click();
- Mouse.end();
- Mouse.move();
- Mouse.press();
- Mouse.release();
- Mouse.isPressed().

Keyboard має функції для роботи з клавіатурою, що дозволяють Leonardo, Micro або Due надсилати підключеному комп'ютеру сигнали натискання клавіш. До бібліотеки Keyboard входять наступні функції:

- Keyboard.begin();
- Keyboard.end();
- Keyboard.press();
- Keyboard.print();
- Keyboard.println();
- Keyboard.release();
- Keyboard.releaseAll();
- Keyboard.write().

2.3.4. Бібліотека SPI

Бібліотека SPI дозволяє контролеру Arduino взаємодіяти з пристроями, що підтримують протокол SPI. Arduino в даному випадку виступає як провідний пристрій.

Послідовний периферійний інтерфейс (SPI) – це послідовний синхронний протокол передачі даних, який використовується мікроконтролерами для обміну даними з одним або декількома периферійними пристроями на невеликих відстанях [1].

Для організації з'єднання SPI необхідний один провідний пристрій, зазвичай це мікроконтролер, який керує з'єднанням з відомими пристроями. Зазвичай підключення здійснюється трьома загальними лініями та лінією вибору периферійного (відомого) пристрою:

- Master In Slave Out (MISO), що перекладається як "вхід ведучого вихід веденого", використовується для передачі даних від веденого до ведучого.

- Master Out Slave In (MOSI) - вихід ведучого вхід веденого для передачі даних від ведучого до периферійних пристроїв.

- Serial Clock (SCK) - синхронізуюча лінія, синхросигнал генерується провідним пристроєм.

- Slave Select pin - вхід на ведених пристроях, за допомогою якого ведучий може ініціювати обмін даними з периферійним пристроєм. Якщо на цьому вході LOW, то ведений взаємодіє з провідним, якщо HIGH, то ведений ігнорує сигнали від ведучого.

2.3.5. Бібліотека Wire

Бібліотека Wire використовується для зв'язку мікроконтролера з пристроями та модулями через інтерфейс I2C. Для зв'язку I2C використовується всього два контакти: лінія даних (SDA) і лінія тактового сигналу (SCL). До відповідних портів Arduino можна підключити до 120 пристроїв, що підтримують інтерфейс I2C. Для обміну даними з такими пристроями і потрібна бібліотека Wire [1].

2.3.6. Бібліотека TimeLib

Ця бібліотека дозволяє Arduino стежити за часом, не використовуючи при цьому стороннє обладнання. Вона дозволяє скетчу отримувати інформацію про поточний час і дату: секунду, хвилину, годину, день, місяць і рік. Крім того, вона використовує тимчасовий тип даних `time_t` (стандартний для мови C), що дозволяє легко розрахувати минулий час і використовувати отримані тимчасові значення на різних платформах.

2.3.7. Бібліотека DS1307RTC

Бібліотека для зручної взаємодії годинника DS1307 з Ардуіно з використанням бібліотеки Wire. Використовуйте мікросхему годинника реального часу DS1307 з бібліотекою часу [11].

2.3.8. Бібліотека MFRC522

Бібліотека необхідна для роботи і взаємодії Arduino з однойменним модулем. До складу бібліотеки входять кілька прикладів, а також компонент Fritzing [13].

2.4. Опис структури системи та алгоритмів її функціонування

На рис. 2.10. зображено блок-схему роботи приладу.

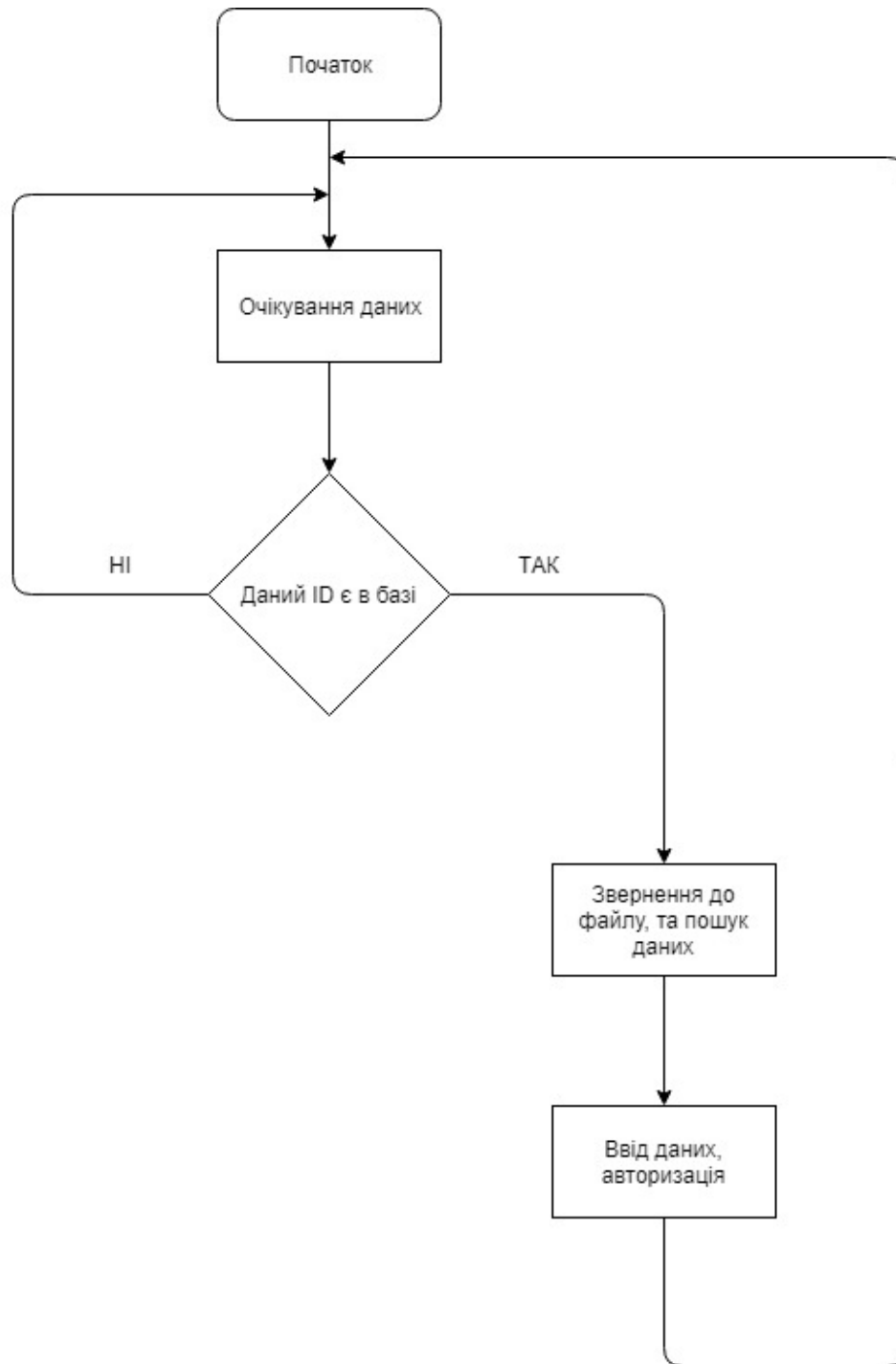


Рис. 2.10. Блок-схема роботи приладу

2.5. Обґрунтування та організація вхідних та вихідних даних

Вхідними даними для розробленого приладу є заздалегідь прописані паролі та ідентифікатори RFID міток та ідентифікатори що зчитують під час його роботи.

Вихідними даними для розробленого приладу у разі збігу сканованого ідентифікатора з записаними є сигнали, що емулюють роботу клавіатури і передають необхідні паролі.

2.6. Опис розробленої інформаційної системи

2.6.1. Використані технічні засоби

Для роботи розробленого приладу можна використовувати будь-який персональний комп'ютер, на якому є доступ до USB портів.

2.6.2. Використані програмні засоби

2.6.2.1. Arduino IDE

Середовище розробки Arduino 1.6.9 представляє собою меню, панель інструментів, текстовий редактор, вікно вводу чи виводу (консоль) та область повідомлень. Зовнішній вигляд програми зображено на рис. 2.11. [3].

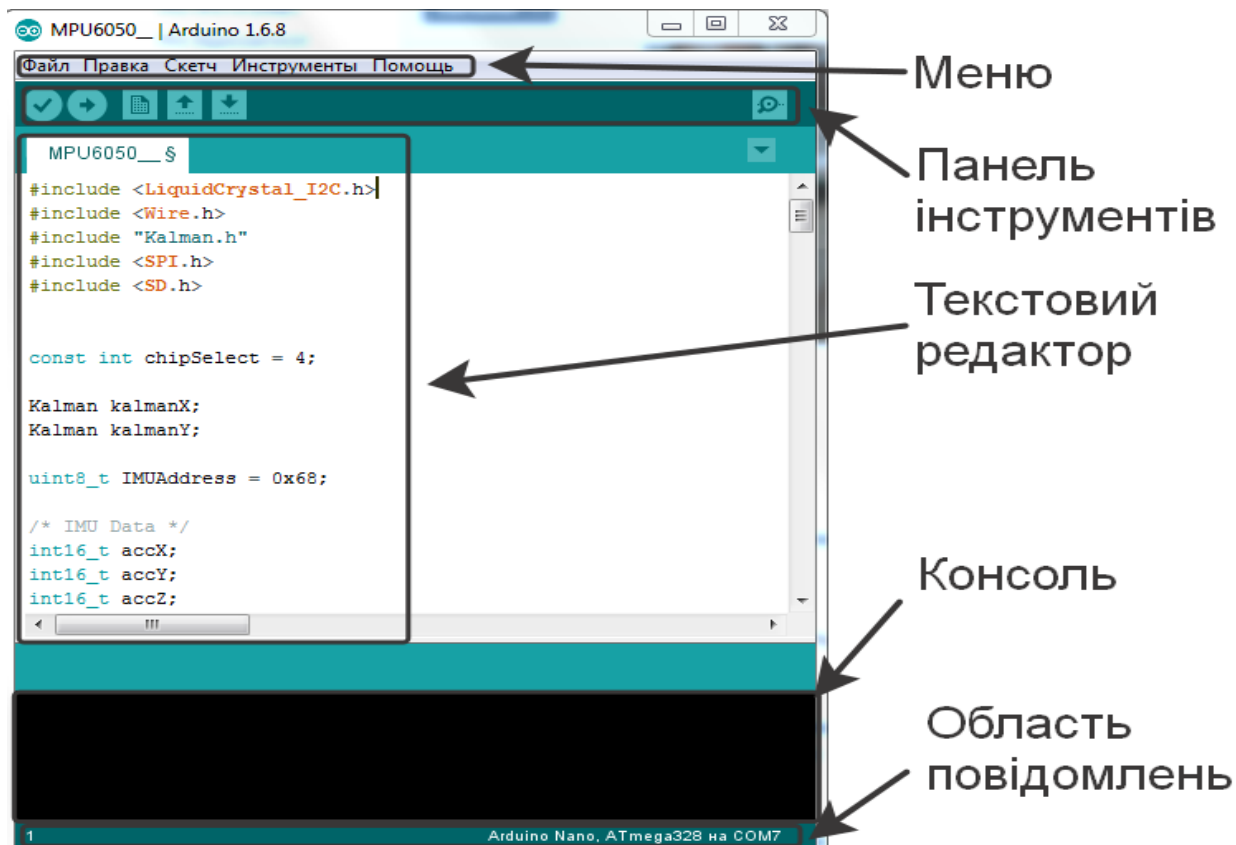


Рис 2.11. Зовнішній вигляд програми

Меню «Файл» в каталозі зі скетчами, по замовчуванню Arduino IDE зберігає кожен скетч в окремий каталог. Ім'я каталогу співпадає з іменем вказаним для скетча при збереженні. Змінити робочу директорію для каталогів зі скетчами можна в пункті меню «Налаштування». Меню «Налаштування» зображено на рис. 2.12.

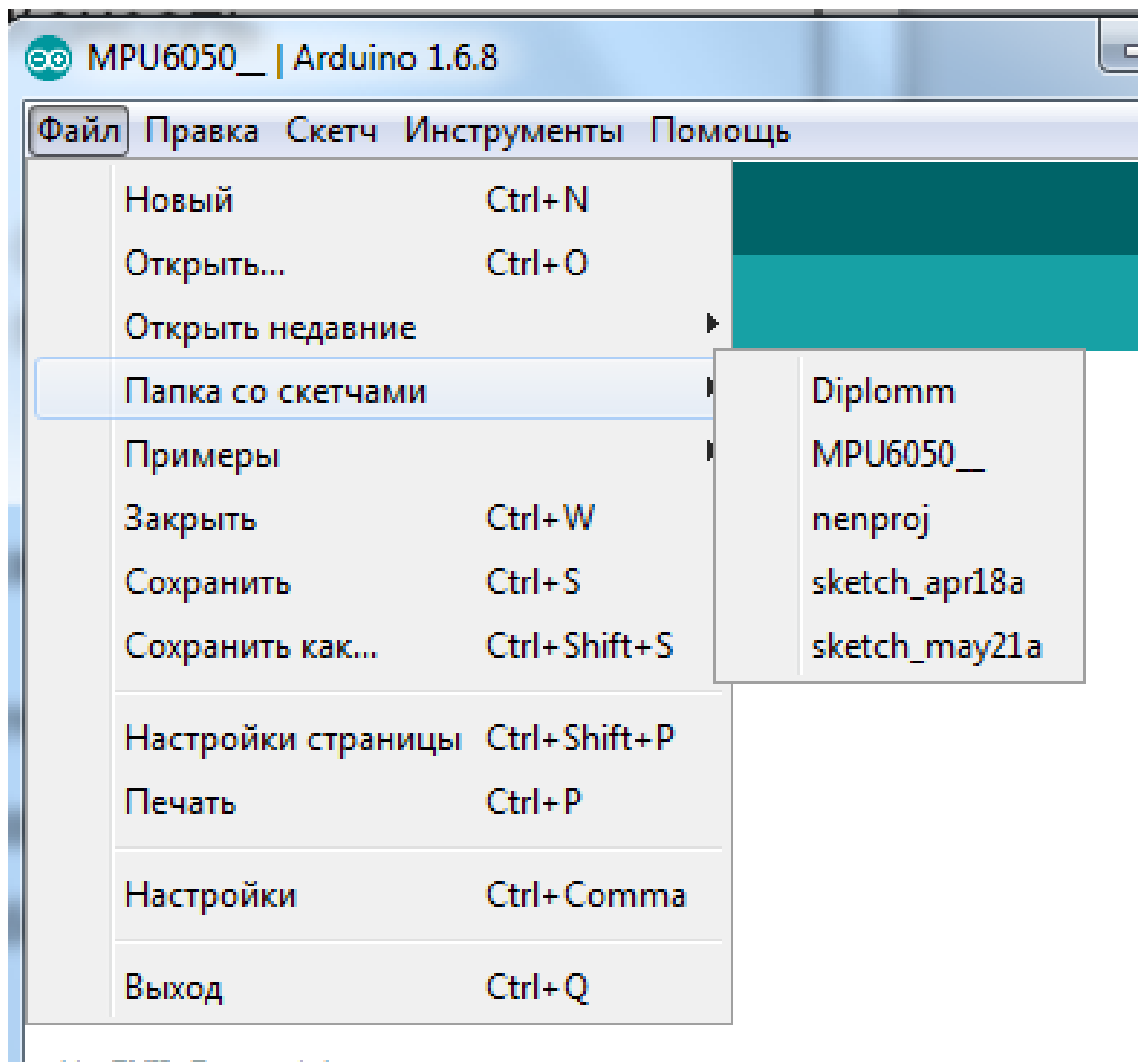


Рис. 2.12. Меню «файл»

В меню "Правка" розміщені команди для роботи з кодом програми. Часто використовувані команди зручно наявністю комбінацій для швидкого доступу за допомогою клавіатури. Зручними функціями являється можливість копіювання для форумів в html форматі. Меню «правка» зображено на рис 2.13.

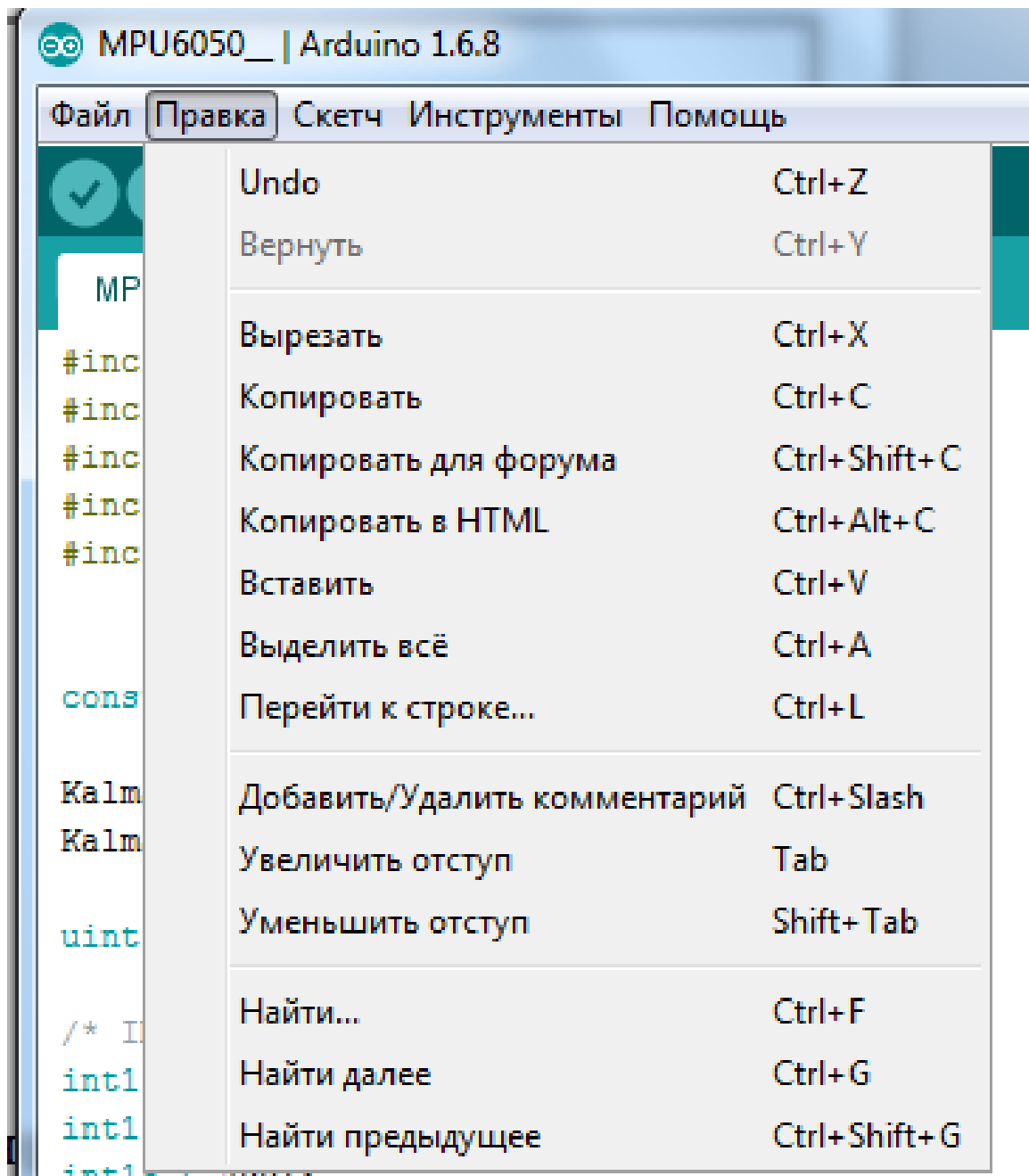


Рис. 2.13. Меню «Правка»

Меню «Скетч». В даному меню продубльована команда з панелі управління «Компілювати». Виконування якої призведе до перевірки коду на помилки, і в випадку їх відсутності – до компіляції. Пункт меню «Показати папку скетчей» відкриє робочу директорію Arduino IDE, вказану в налаштуваннях. «Добавить файл...» дозволяє відкрити текстовий файл в іншій вкладці. Рядок «Добавить файл» зображена на рис. 2.14. [3].

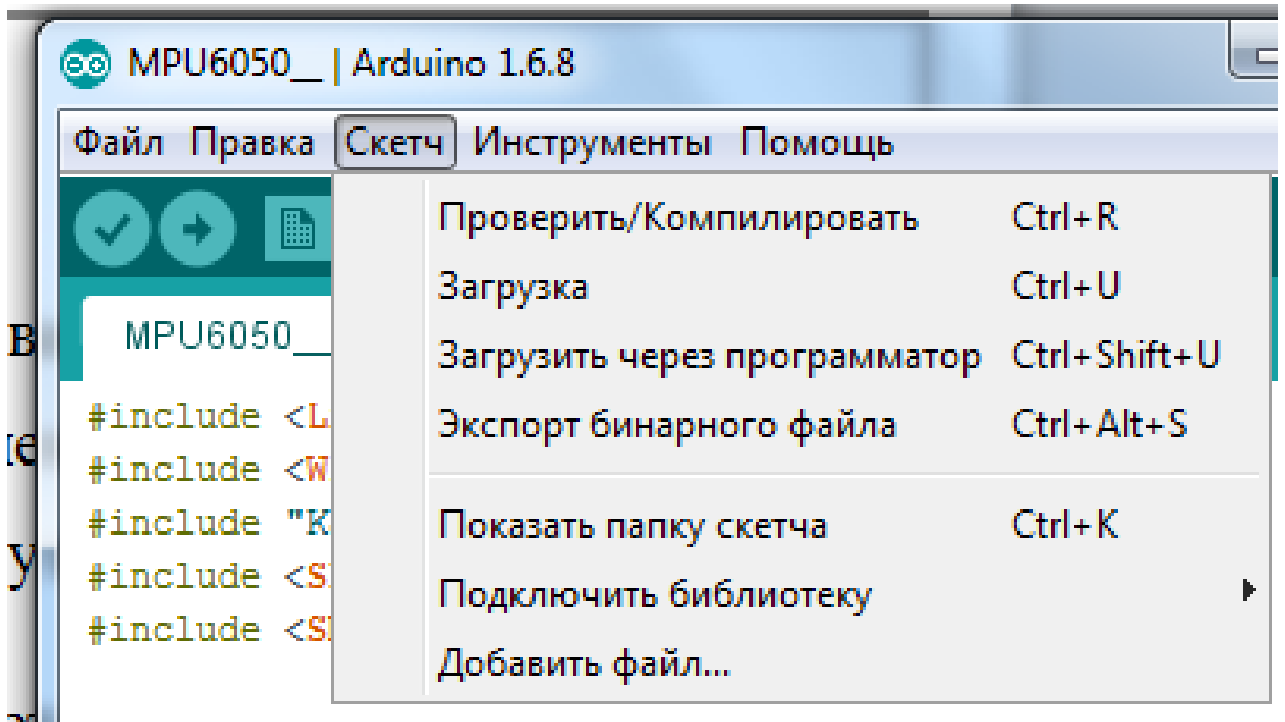


Рис. 2.14. Меню «Скетч»

Бібліотеки додають додаткову функціональність скетчам, наприклад при роботі з апаратною частиною або при обробці даних. Одна чи декілька директив `#include` будуть розміщені на початку коду скетча з подальшою компіляцією бібліотек і разом зі скетчем. Загрузка бібліотек потребує додаткового місця в пам'яті Arduino. Для встановлення бібліотек можна скористатися командою «Импортировать библиотеку...». Команда «Импортировать библиотеку...» зображена на рис 2.15.

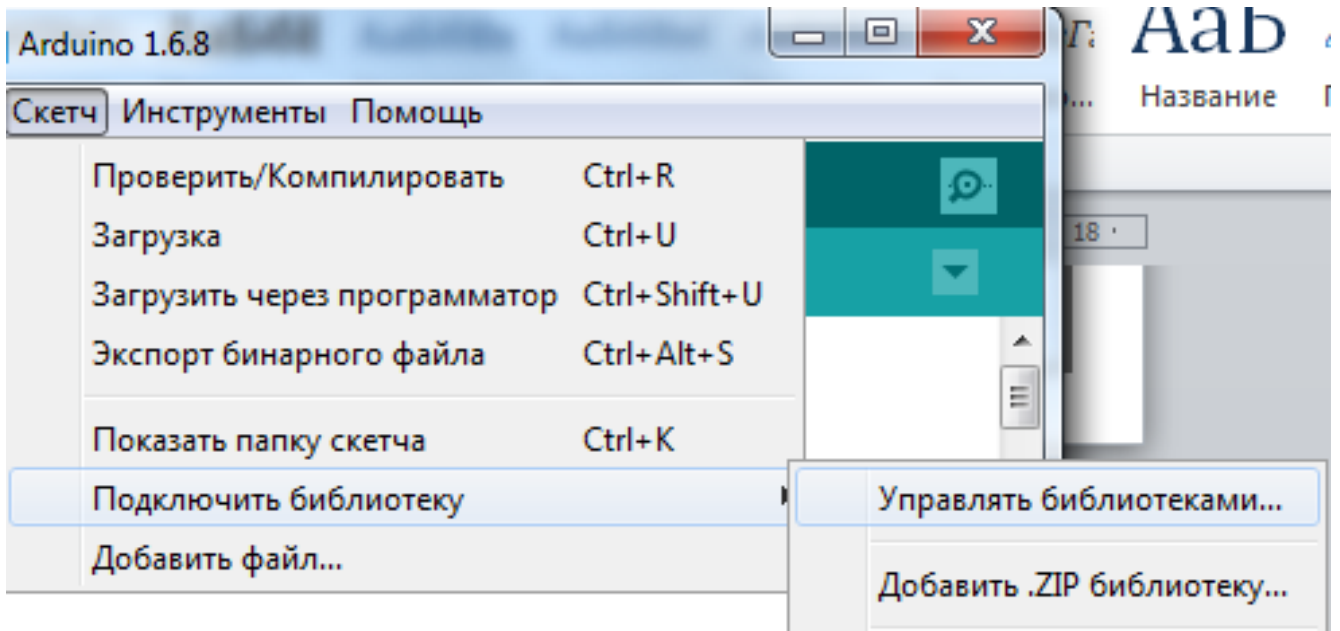


Рис. 2.15. Команда «Импортировать библиотеку...»

Після додавання бібліотеки може бути необхідно перезавантаження середовища програмування. Пункт меню «Монитор порта» викликає вікно для обміну з повідомленнями контролером через СОМ порт. Вікно для обміну повідомленнями зображено на рис 2.16.

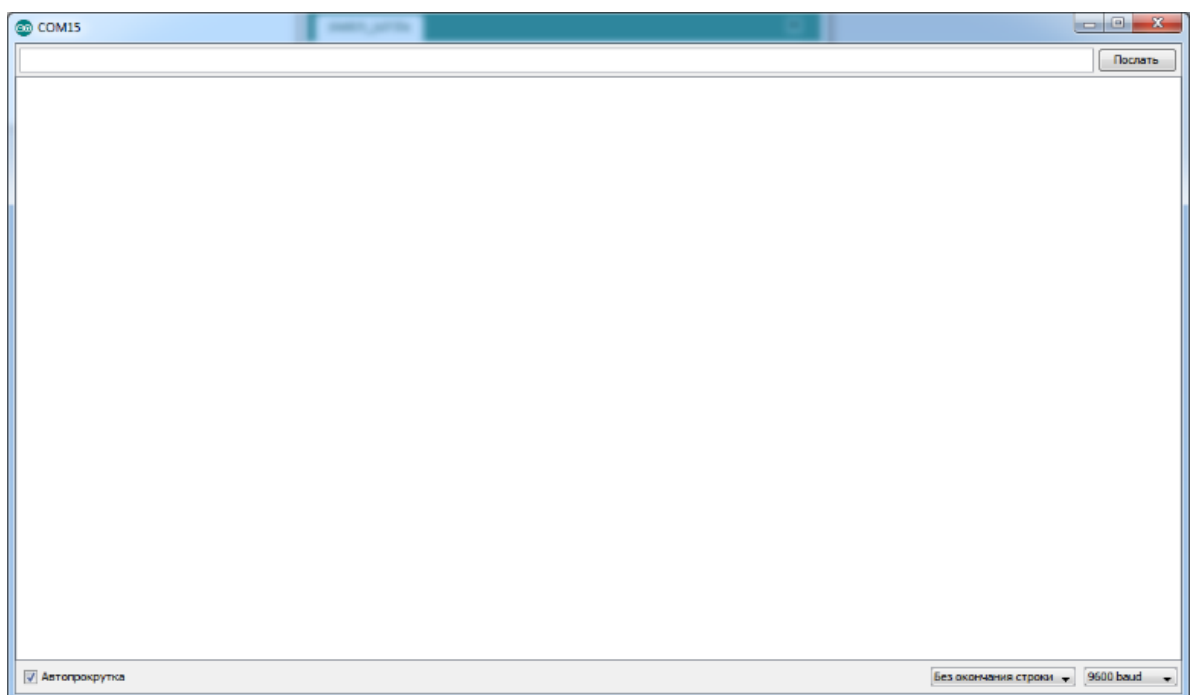


Рис 2.16. Вікно для обміну повідомленнями.

Меню «Інструменти». В даному меню знаходяться налаштування які вказують на підключений пристрій, присутній пункт вікна обміну повідомленнями, та різні операції зі скетчами. Меню «Інструменти» зображено на рис. 2.17.

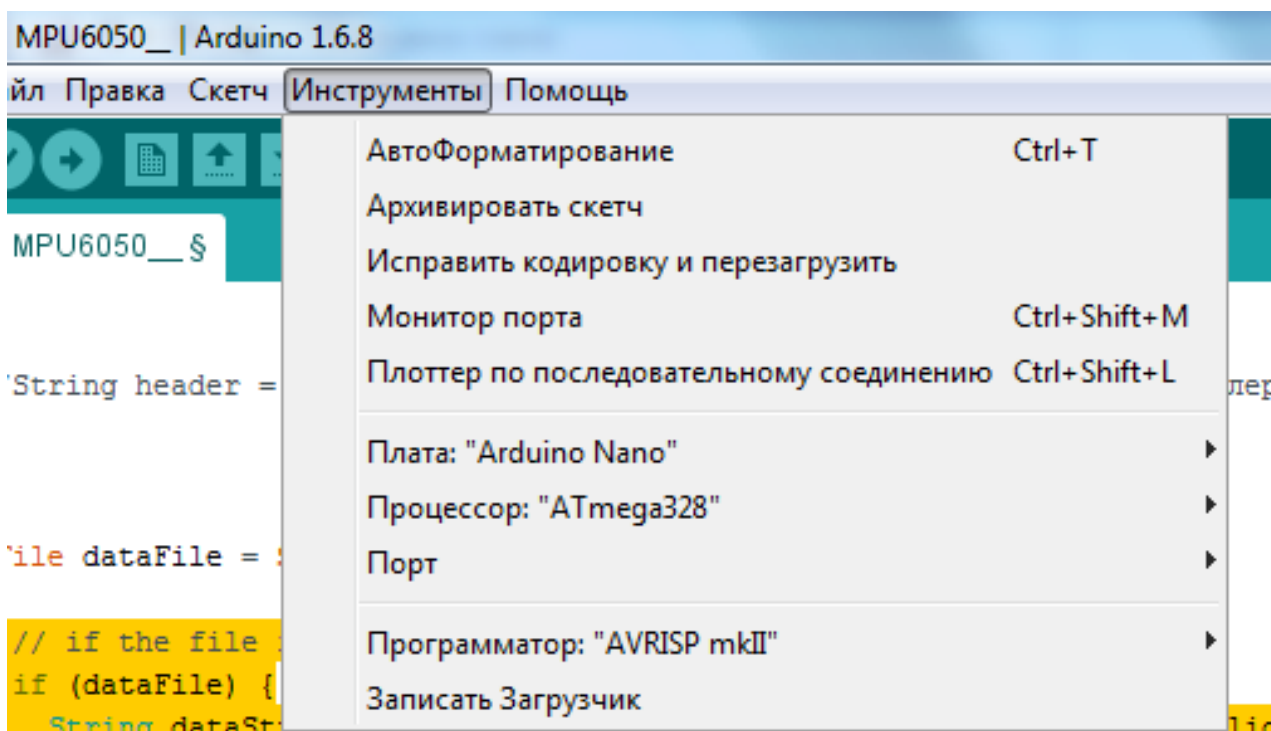


Рис 2.17. Меню «Інструменти»

Загалом середовище розробки має переваги в своєму інтерфейсі, так як інтерфейс максимально зрозумілий та зручний для користувача.

2.6.2.2. Fritzing

Програмний пакет Fritzing може стати в нагоді в таких стадіях розробки, як малюнок прототипу схеми на макетній платі, а також автоматичне генерування важливої схеми та друкованої плати. Цільова аудиторія програми – творчі люди, дослідники, дизайнери, радіоаматори, які працюють із інтерактивними електричними пристроями. Fritzing створювалася для Arduino, апаратно-програмної платформи, що складається із звичайної плати з мікроконтролером Atmel AVR, радіодеталей для програмування, інтерфейсів зв'язку, середовища розробки Processing/Wiring. Arduino використовується у створенні автономних інтерактивних апаратів. Малюнок друкованої плати відкритий всім, і тому платформа поширена серед самостійних розробників. Зовнішній вигляд робочих вікон зображено на рис. 2.18-2.20. [20].

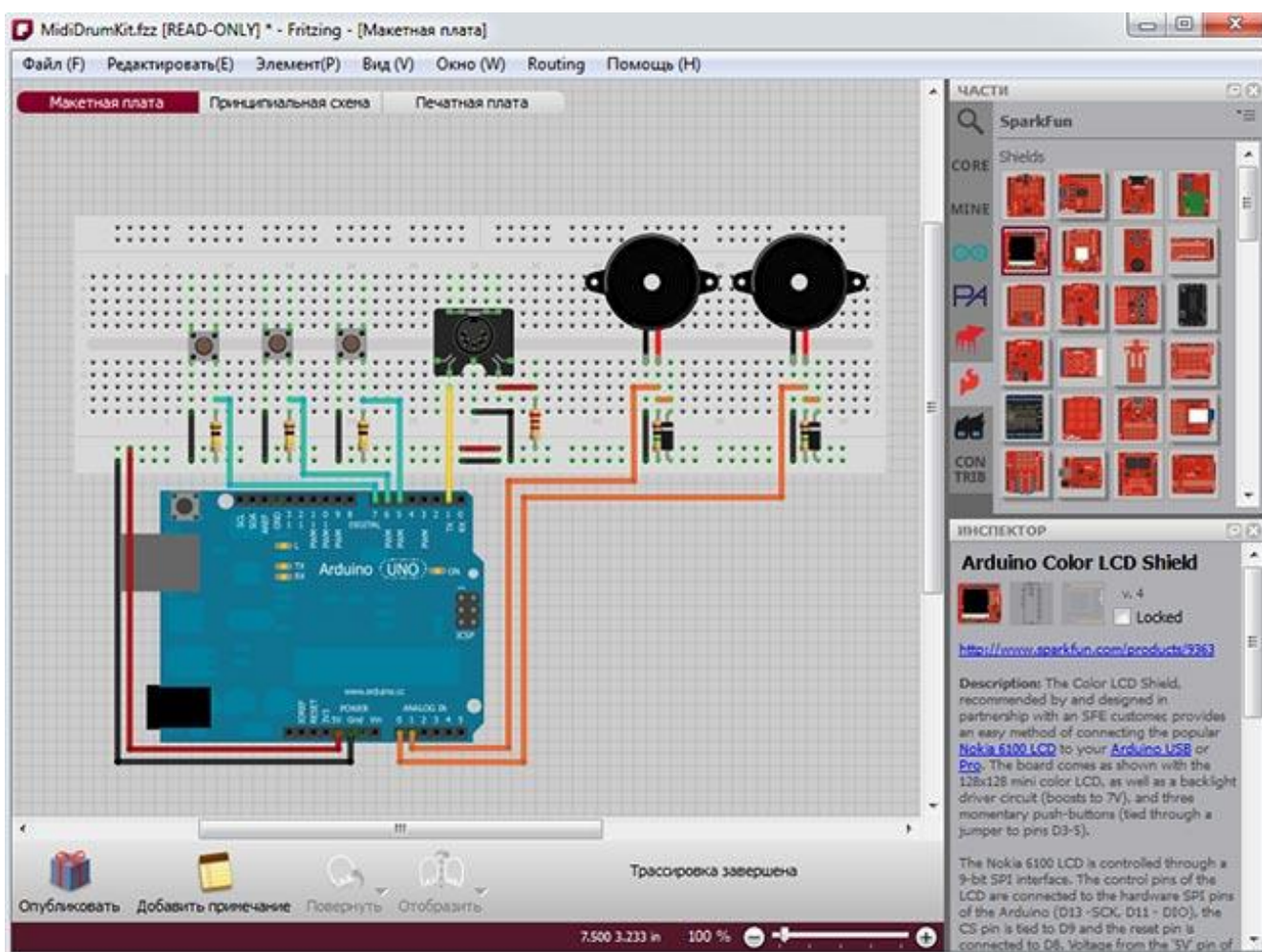


Рис 2.18. Робоче вікно програми Fritzing (макетна плата)

Робота з новим проектом у пакеті Fritzing починається з вибору готових компонентів, повний перелік яких розташований у верхньому куті робочого вікна з правого боку. Тут можна знайти різні макетні та монтажні плати (у тому числі Arduino), цілий набір аналогових та цифрових мікросхем, будь-які радіодеталі: конденсатори, транзистори, резистори, світлодіоди, батарейки, кнопки. Додатково є велика колекція пристроїв для робототехніки: мотори, далекоміри, динаміки, «пищалки», сервоприводи, крокові двигуни, LCD та цифрові індикатори, а також багато іншого. Також можна створювати власні елементи та оновлювати існуючу базу. Схема доступна для малювання як у вікні «Макетна плата», так і у вікні «Принципова схема» простим перетягуванням потрібних компонентів на робоче поле. Наявність є функція автотрасування. При виборі вікна «Друкована плата» можна розпочати розведення провідників та розміщення елементів. Результат роботи експортується у pdf-файл для роздруківки на принтері.

На жаль, самостійно розводити друковані плати Fritzing не може, як і симулювати роботу схем. Малювання спрощено, а елементна база дуже мала. Красиві, яскраво-барвисті, детальні схеми, великі схеми більше підходять для фотозвітів та швидких нарисів, що пояснюють обґрунтування роботи того чи іншого пристрою.

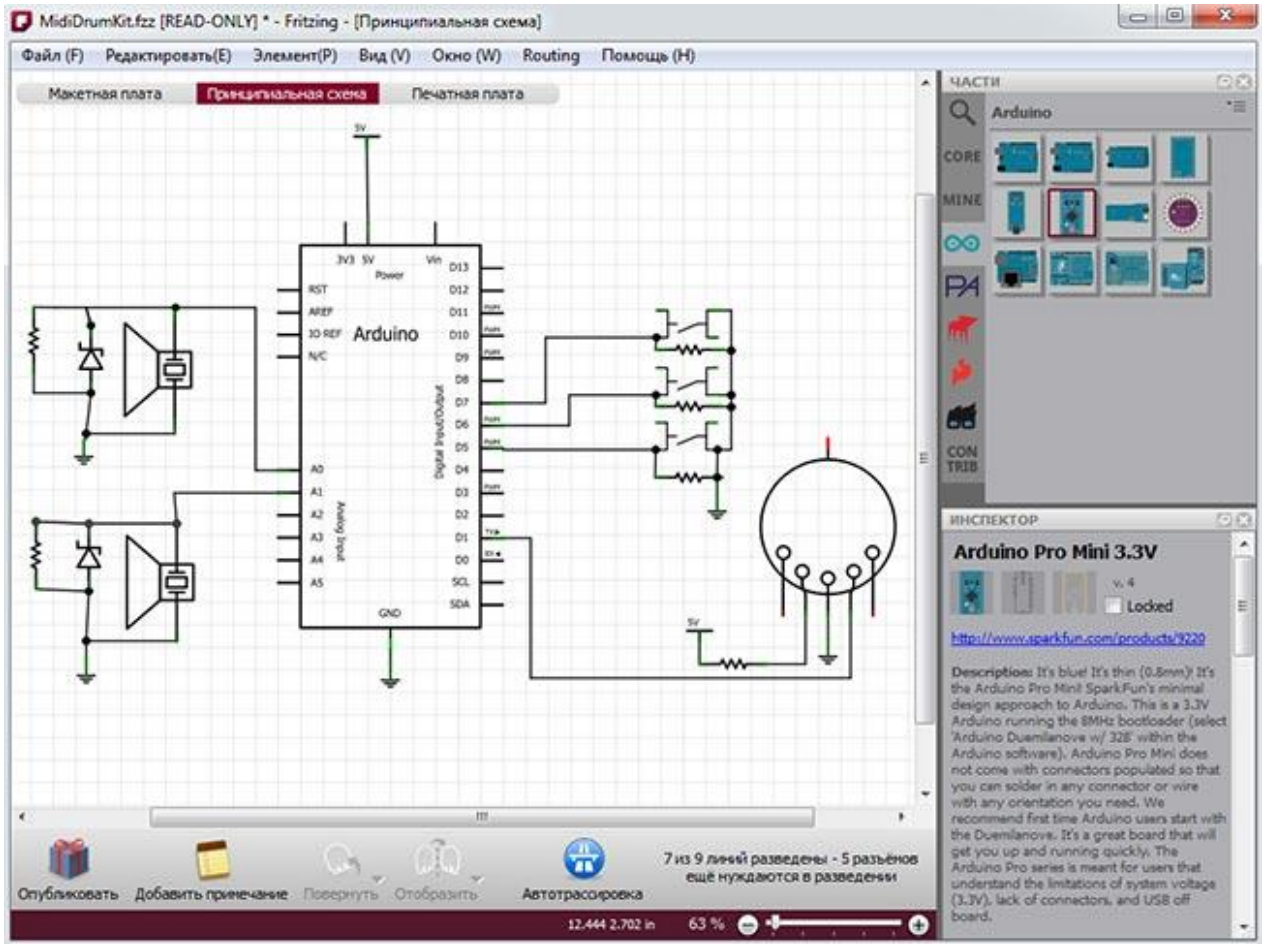


Рис. 2.19. Рабочее окно программы Fritzing (принципова схема)

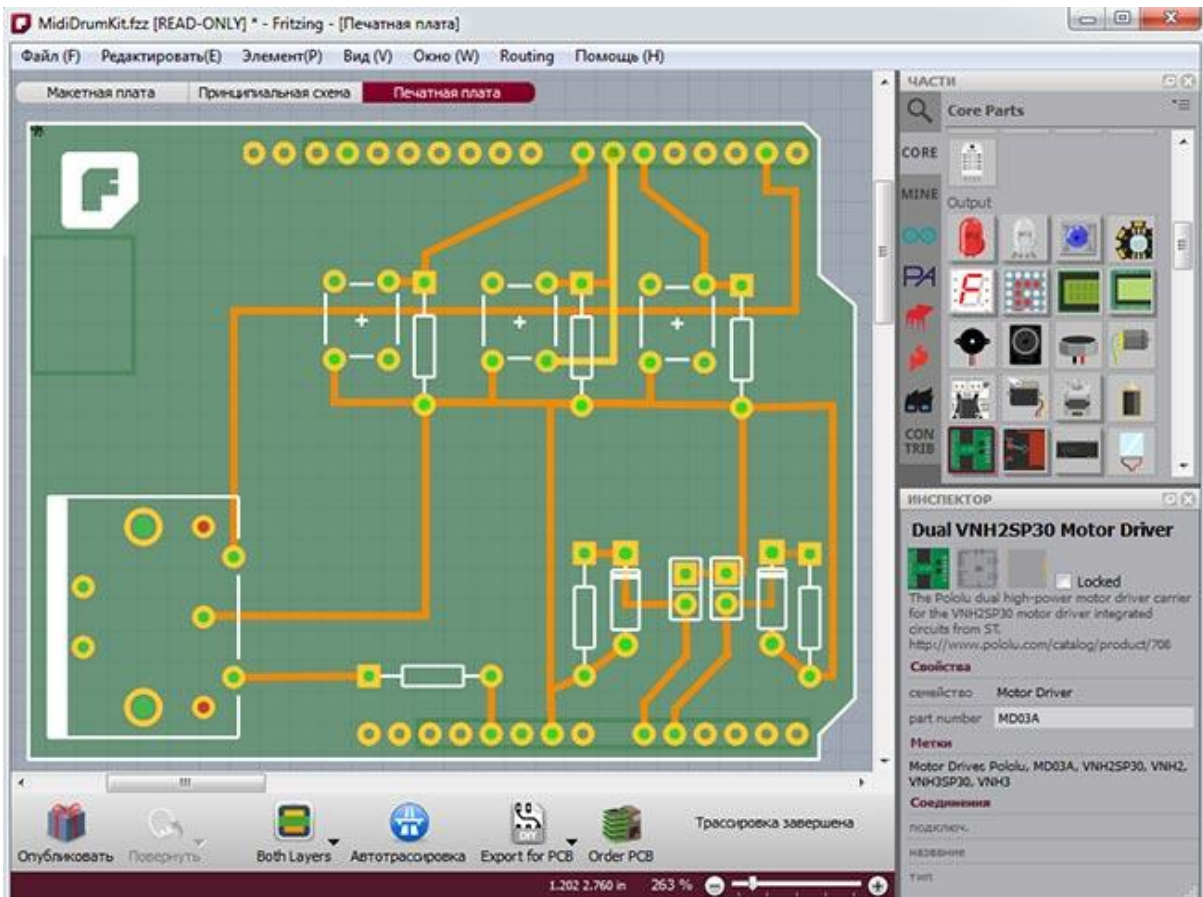


Рис. 2.20. Рабочее окно программы Fritzing (друкована плата)

2.6.3. Виклик та завантаження інформаційної системи

Завантаження та настройка розробленого приладу реалізується безпосередньо на моменті запису скетчу у самому кодї після чого не може змінюватися та зчитуватися. Для змін основних параметрів необхідно перепрошивати пристрій повторно.

Підключення налаштованого приладу полягає лише у підключенні приладу до необхідного ПК через USB порт.

Для початку використання пристрій необхідно підключити за допомогою USB. Після цього необхідно дізнатися до якого з COM – портів підключений прилад. COM порт — двонаправлений послідовний інтерфейс, призначений для обміну байтовою інформацією. Послідовний тому, що інформація через нього передається по одному біту, біт за бітом. Найчастіше для послідовного порту персональних комп'ютерів використовується стандарт RS-232c.

Для визначення номера порта треба перейти до управління комп'ютером, а потім до диспетчера приладів, де у графі COM портів відобразиться прилад зображено на рис 2.21.

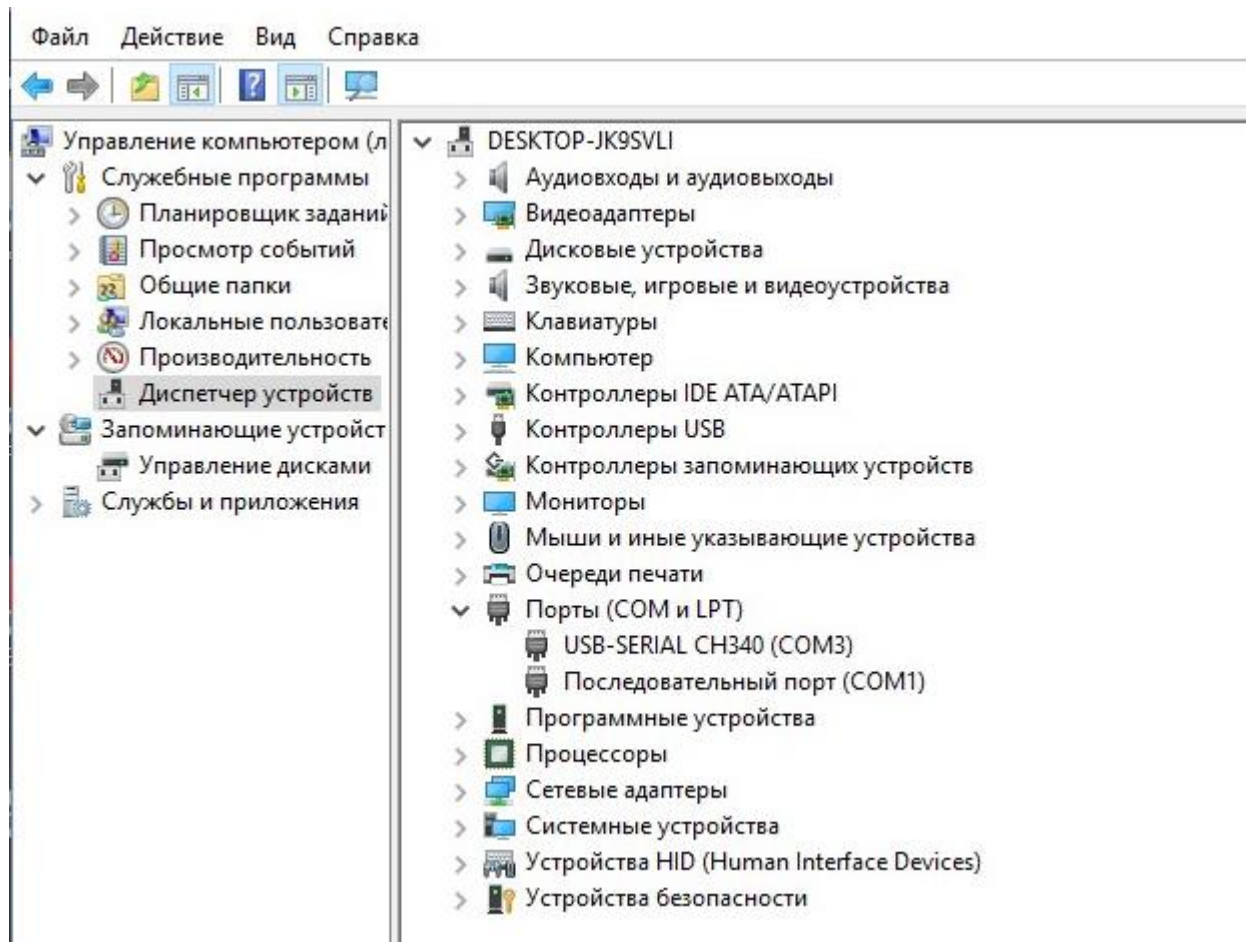


Рис 2.21. Відображення приладу у порті COM

На даний момент прилад підключено до порту COM 3, що треба зазначити у налаштуванні. Після цього треба відкрити вікно порта за зчитати ID ключа який буде використовуватися. Для цього відкриваємо програму Arduino IDE та переходимо до вкладки “Инструменты” а потім ”Монитор порта” зображено на рис. 2.22.

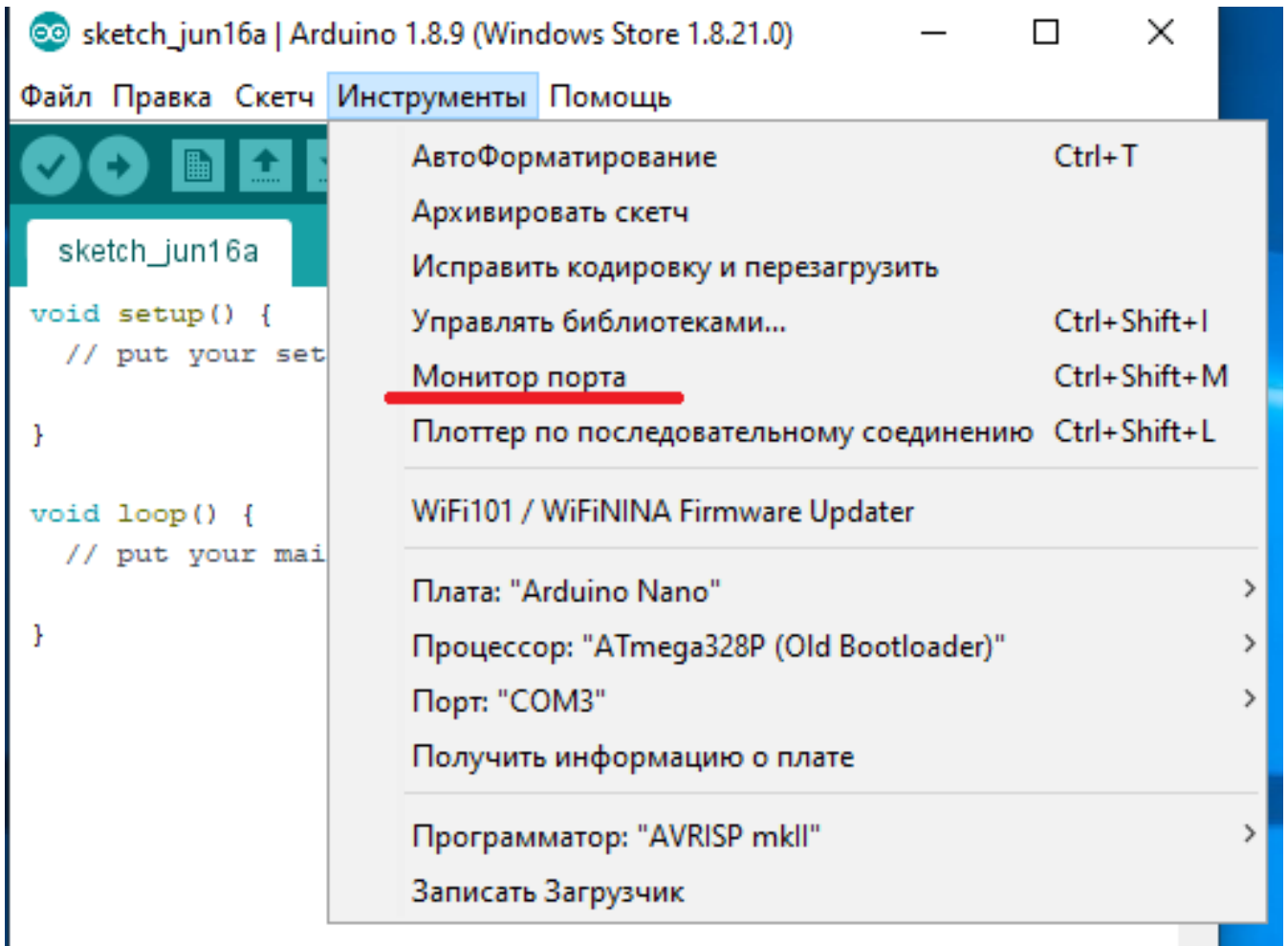


Рис 2.22. Зображення меню інструменти

Після відкриття монітору порта треба піднести до приладу свій ключ, і ми отримаємо його ID, рис. 2.23.

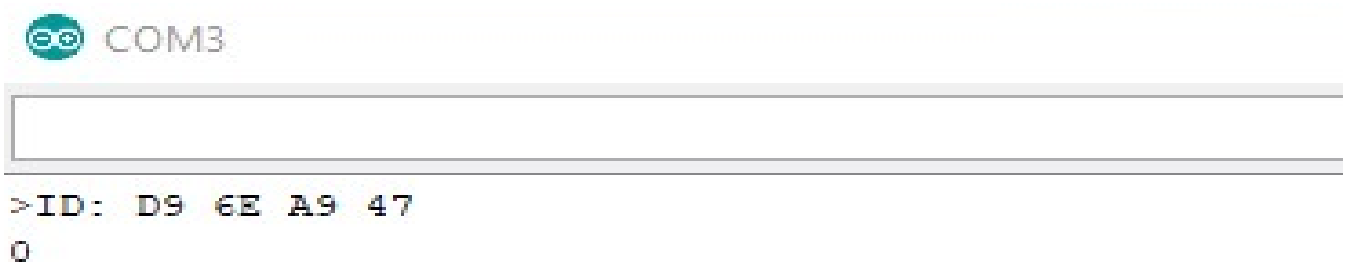


Рис. 2.23. Монітор порта

Після цього ми записуємо до відповідного ID пароль користувача, і при зчитуванні цього ID прилад вибере заданого користувача і введе пароль.

2.6.4. Опис інтерфейсу користувача

Інформаційна система що розробляється є програмно-апаратним рішенням, що слугує допоміжним засобом для входу в інтерфейс операційної системи, тому власного графічного інтерфейсу не має.

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки інформаційної системи

Початкові дані:

1. передбачуване число операторів інформаційної системи– 100;
2. коефіцієнт корекції програми в ході її розробки – 0,05;
3. коефіцієнт складності програми – 1,7;
4. годинна заробітна плата програміста– 135 грн/год;

Станом на кінець 2021 року зарплата Arduino розробника простягається від 500\$ до 1100\$. Вирахувавши середню заробітну плату програміста маємо плату 800\$ у місяць. При курсі валют НБУ на початок червня 2022 року один американський долар дорівнює 29,76 грн, тому середня зарплата в гривнях дорівнює 23800 грн. При стандартному графіку (176 годин/місяць) зарплата за годину буде становити близько 135 грн. [10].

5. коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі – 1,4;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,4;
7. вартість машино-години ЕОМ –21 грн/год.

Для цього проекту потрібна не дуже велика потужність ПК, проте якщо його немає на час розробки гарним рішенням буде оренда комп'ютера. Вартість оренди комп'ютера на місяць 1300 грн (монітор) та 2400 грн (системний блок). Загалом на місяць оренда коштуватиме 3700 грн. При стандартному графіку (176 годин/місяць) вартість машино-години ЕОМ за годину роботи буде становити 21 грн. В цю вартість входить ремонт за гарантією та базовий комплект гарнітури (клавіатура та миша).

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки інформаційної системи можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_{\partial}, \text{ людино-годин, (3.1)}$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n -витрати праці на програмування по готовій блок-схемі та монтажу приладу;

t_{oml} -витрати праці на налагодження інформаційної системи на ЕОМ;

t_{∂} - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у інформаційної системи, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p),$$

де q - передбачуване число операторів (3050);

C - коефіцієнт складності інформаційної системи (1,7);

p - коефіцієнт корекції інформаційної системи в ході її розробки (0,05).

Звідси умовне число операторів інформаційної системи:

$$Q = 1,7 \cdot 100 \cdot (1 + 0,05) = 178,5$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням

уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин,}$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 5 до 8 років він складає 1,4.

Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1,2$). З урахуванням коефіцієнта кваліфікації $k = 1,4$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (178.5 \cdot 1,2) / (75 \cdot 1,4) = 2,04 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин, (3.2)}$$

де Q – умовне число операторів інформаційної системи;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.2), отримаємо:

$$t_a = 178,5 / (20 \cdot 1,4) = 6,375 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі та монтажу інформаційної системи:

$$t_n = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин.}$$

$$t_n = 178,5 / (25 \cdot 1,4) = 5,1 \text{ людино-годин.}$$

Витрати праці на налагодження інформаційної системи на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин.}$$

$$t_{oml} = 178,5 / (5 \cdot 1,4) = 25,5 \text{ людино-годин.}$$

- за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин.}$$

$$t_{oml}^k = 1,5 \cdot 25,5 = 38,25 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\delta} = t_{\delta p} + t_{\delta o}, \text{ людино-годин,}$$

де $t_{\delta p}$ -трудомісткість підготовки матеріалів і рукопису:

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k}, \text{ люДИНО-ГОДИН,}$$

$t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації:

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ люДИНО-ГОДИН.}$$

Підставляючи відповідні значення, отримаємо:

$$t_{\partial p} = 178,5 / (18 \cdot 1,4) = 7,1 \text{ люДИНО-ГОДИН.}$$

$$t_{\partial o} = 0,75 \cdot 7,1 = 5,325 \text{ люДИНО-ГОДИН.}$$

$$t_{\partial} = 7,1 + 5,325 = 12,425 \text{ люДИНО-ГОДИН.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 2,04 + 6,375 + 5,4 + 25,5 + 12,425 = 101,74 \text{ люДИНО-ГОДИН.}$$

3.2. Розрахунок витрат на створення інформаційної системи

Витрати на створення інформаційної системи $K_{\text{ПО}}$ включають витрати на заробітну плату виконавця програми $Z_{\text{ЗП}}$ і витрат машинного часу, необхідного на налагодження інформаційної системи на ЕОМ:

$$K_{\text{ПО}} = Z_{\text{ЗП}} + Z_{\text{МВ}}, \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{зп} = t \cdot C_{пр}, \text{ грн,}$$

де: t - загальна трудомісткість, людино-годин;

$C_{пр}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 135 грн / год, отримуємо:

$$Z_{зп} = 101,74 \cdot 135 = 13\,735 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження інформаційної системи на ЕОМ, визначається за формулою:

$$Z_{мв} = t_{отл} \cdot C_{мч}, \text{ грн, (3.3)}$$

де $t_{отл}$ - трудомісткість налагодження інформаційної системи на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (21 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{мв} = 25,5 \cdot 21 = 535,5 \text{ грн.}$$

Звідси витрати на створення інформаційної системи:

$$K_{по} = 13\,735 + 535,5 = 14\,270,5 \text{ грн.}$$

Очікуваний період створення інформаційної системи:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.}$$

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Звідси витрати на створення інформаційної системи:

$$T = 101,74 / 1 \cdot 176 \approx 0,57 \text{ міс.}$$

Висновок: інформаційна система розроблена для забезпечення легкого доступу користувачів до ПК. Вартість даної інформаційної системи 14 270,5 грн і не вимагає додаткових витрат як при розробці програми. Очікуваний час розробки становить 101,74 годин, тобто 0,57 місяців. Цей термін пов'язаний зі числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження приладу, монтажу і підготовку документації.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було поставлено завдання розробити апаратний логгер на базі МК Arduino.

Ця інформаційна система призначена для надання доступу до ПК з використанням унікальних RFID міток. Практичне призначення даної інформаційної системи полягає в захищеному та полегшеному доступу користувачів до робочих ПК за умови, що самі робітники можуть навіть і не знати робочих паролів, що виключає можливість зливів паролів шляхом соціальної інженерії та ускладнює корпоративний шпіонаж.

Під час виконання даного проекту були виконані наступні задачі:

- вивчено предметну галузь розв'язуваної задачі;
- створено алгоритм для реалізації поставленого завдання;
- створено інформаційну систему;
- легке та доступне адміністрування інформаційної системи;
- формування звітних записів щодо часу та користувача, що взаємодіє з інформаційною системою.

Програма інформаційної системи реалізована на базі Arduino IDE. З використанням мови програмування Arduino C, що представляє собою мову C++ з використанням фреймворку Wiring та має деякі відмінності від стандартного C++.

Прилад реалізований на базі МК Arduino Leonardo з використанням сумісних модулів.

Також у кваліфікаційній роботі було визначено трудомісткість розробленої інформаційної системи, на базі середньої зарплати розробника проведений підрахунок вартості роботи по створенню програми, який складає 14 270,5 грн та розраховано час на створення – 101,74 людино-годин , тобто 0,57 місяців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стив Логан Управління кількома периферійними пристроями по лініях шин SPI та I2C.
2. Александр Белов ARDUINO. От азов до создания практических устройств.
3. Аджиев Р.А. Картавцев Д.В. Мікроконтролери. Arduino IDE середовище розробки.
4. Офіційний сайт Arduino [Електроний ресурс] - Режим доступу: <https://www.arduino.cc/>.
5. Вартість аренды ноутбуку почасово [Електроний ресурс] - Режим доступу: <https://notebooksbu.com/garantiya-3-goda/>.
6. Офіційний магазин Arduino. Arduino Leonardo [Електроний ресурс] - Режим доступу: <https://store.arduino.cc/products/arduino-leonardo-with-headers>.
7. RFID модуль RC522 [Електроний ресурс] - Режим доступу: <https://arduino.ua/prod649-rfid-modyl-rc522-s-kartochkoi-dostupa-dlya-arduino>.
8. Модуль microSD карти [Електроний ресурс] - Режим доступу: <https://arduino.ua/prod1601-modyl-micro-sdtf-card>.
9. RTC модуль на базі DS1307 [Електроний ресурс] - Режим доступу: <https://arduino.ua/prod724-Real-Time-Clock-Modyl-DS1307-bez-batareiki>.
10. Зарплати українських розробників - літо 2021 [Електроний ресурс] - Режим доступу: <https://dou.ua/lenta/articles/salary-report-devs-summer-2021/>.
11. Github DS1307RTC [Електроний ресурс] - Режим доступу: <https://github.com/PaulStoffregen/DS1307RTC>.
12. Github Arduino RC522 RFID Access Control [Електроний ресурс] - Режим доступу: <https://github.com/omersiar/RFID522-Door-Unlock>.
13. Github бібліотека MFRC522 [Електроний ресурс] - Режим доступу: <https://github.com/miguelbalboa/rfid>.
14. Atmel ATmega16U4/ATmega32U4 Datasheet.

15. MFRC522 Standard performance MIFARE and NTAG frontend Rev. 3.9 — 27 April 2016.
16. SN74LVC125A Quadruple Bus Buffer Gate With 3-State Outputs SCAS290Q –JANUARY 1993–REVISED JANUARY 2015.
17. DS1307 Datasheet - Dallas Semiconductor.
18. Офіційна база документації Arduino [Електроний ресурс] - Режим доступу: <https://docs.arduino.cc/hardware/leonardo>.
19. RFID-технологія: що це таке. Використання RFID [Електроний ресурс] - Режим доступу: <https://www.1cbit.ru/blog/rfid-tekhnologiya-chto-eto-takoe/>.
20. Офіційний сайт Fritzing [Електроний ресурс] - Режим доступу: <https://fritzing.org/>.

КОД ПРОГРАМИ

```
// Config
#define SD_ENABLED 1
#define USE_SDFAT_LIBRARY 1

// Pins
#define SD_CS_PIN 4
#define MRF_CS_PIN 10
#define MRF_RST_PIN 9

// Libraries
#include <SPI.h>
#include <MFRC522.h>
#include <Keyboard.h>
#include <Wire.h>
#include <TimeLib.h>
#include <DS1307RTC.h>
#if defined(SD_ENABLED)
    #if !defined(USE_SDFAT_LIBRARY)
        #include <SD.h>
    #else
        #include "SdFat.h"
    #endif
#endif
char lang = 'E';
uint32_t user = 0xEC783823;
char password[] = "10012000r";
uint32_t admin = 0xC6B8532B;
char password2[] = "admin12345";
MFRC522 mfrc522(MRF_CS_PIN, MRF_RST_PIN);
#if defined(USE_SDFAT_LIBRARY) && defined(SD_ENABLED)
    SdFat SD;
#endif
#if defined(SD_ENABLED)
    File myFile;
    String sfilename;
```

```

    char filename[20];
#endif
uint32_t id = 0x00000000;
tmElements_t tm;
unsigned long millis1 = 0;
String record = "";
String curr = "";
void setup() {
    Serial.begin(115200);
    Keyboard.begin();
    SPI.begin();
    mfrc522.PCD_Init();
    #if defined(SD_ENABLED)
        SD.begin(SD_CS_PIN);
    #else
        pinMode(SD_CS_PIN, OUTPUT);
        digitalWrite(SD_CS_PIN, HIGH);
    #endif
}
void loop() {
    if ( ! mfrc522.PICC_IsNewCardPresent()) {
        return;
    }
    if ( ! mfrc522.PICC_ReadCardSerial()) {
        return;
    }
    id = 0;
    id = 0x1000000 * mfrc522.uid.uidByte[0] + 0x10000 * mfrc522.uid.uidByte[1] + 0x100 *
mfrc522.uid.uidByte[2] + mfrc522.uid.uidByte[3];
    //id = (mfrc522.uid.uidByte[0] << )
    Serial.println(id, HEX);
    if (id == user)
    {
        curr = "User ";
        pass(password);
    }
}

```

```

}
else if (id == admin)
{
    curr = "Admin ";
    pass(password2);
}
delay(1000);
}
void pass(const char *p)
{
    if (lang == 'R')
    {
        Keyboard.press(KEY_LEFT_ALT);
        Keyboard.press(KEY_LEFT_SHIFT);
        Keyboard.release(KEY_LEFT_SHIFT);
        Keyboard.release(KEY_LEFT_ALT);
        delay(20);
    }
    Keyboard.press(KEY_LEFT_GUI);
    Keyboard.press('l');
    Keyboard.releaseAll();
    delay (100);
    //Keyboard.write(KEY_RETURN);
    //Keyboard.releaseAll();
    delay(300);
    Keyboard.print(p);
    Keyboard.releaseAll();
    delay(500);
    Keyboard.write(KEY_RETURN);
    Keyboard.releaseAll();
    delay(1000);
    #if defined(SD_ENABLED)
        rec(curr);
    #endif
}

```

```

#if defined(SD_ENABLED)
void rec (String curr1) {
  sfilename = get_file_name();
  Serial.print("filename=");
  Serial.println(sfilename+".txt");
  sfilename.toCharArray(filename, 20);
  myFile = SD.open(filename, FILE_WRITE);
  record = get_time();
  record += " ";
  record += String(curr1);
  record += "\n";
  Serial.print("record=");
  Serial.println(record);
  myFile.println(record);
  myFile.close();
}

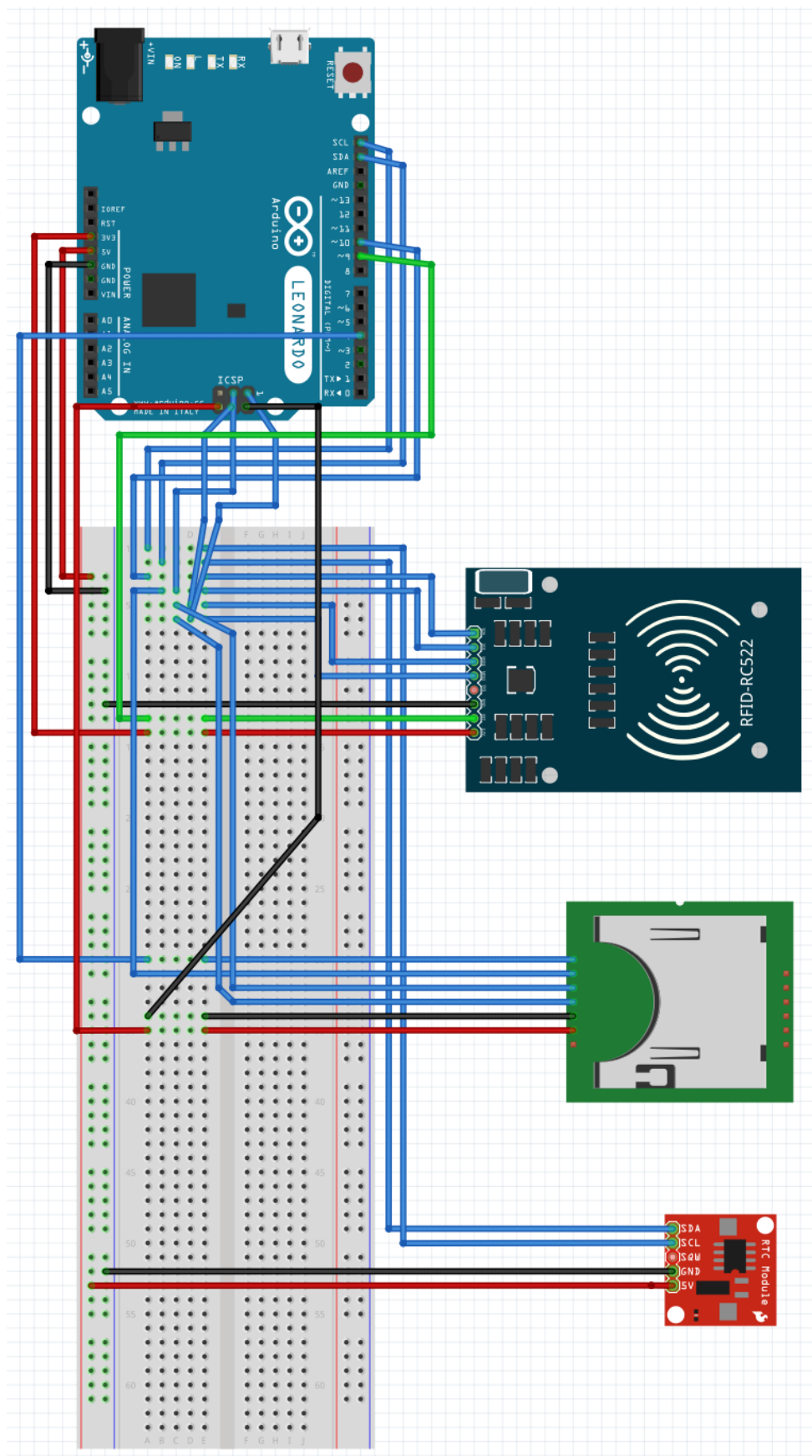
String get_time() {
  String time1;
  RTC.read(tm);
  if (tm.Hour<10){
    time1= "0" + String(tm.Hour, DEC);
  }
  else time1 = String(tm.Hour, DEC);
  if (tm.Minute<10){
    time1 = time1+ ":0" + String(tm.Minute, DEC);
  }
  else time1 = time1+ ":" + String(tm.Minute, DEC);
  return time1;
}

String get_file_name() {
  String date1;
  RTC.read(tm);
  if (tm.Day<10){
    date1= "0" + String(tm.Day, DEC);
  }
}

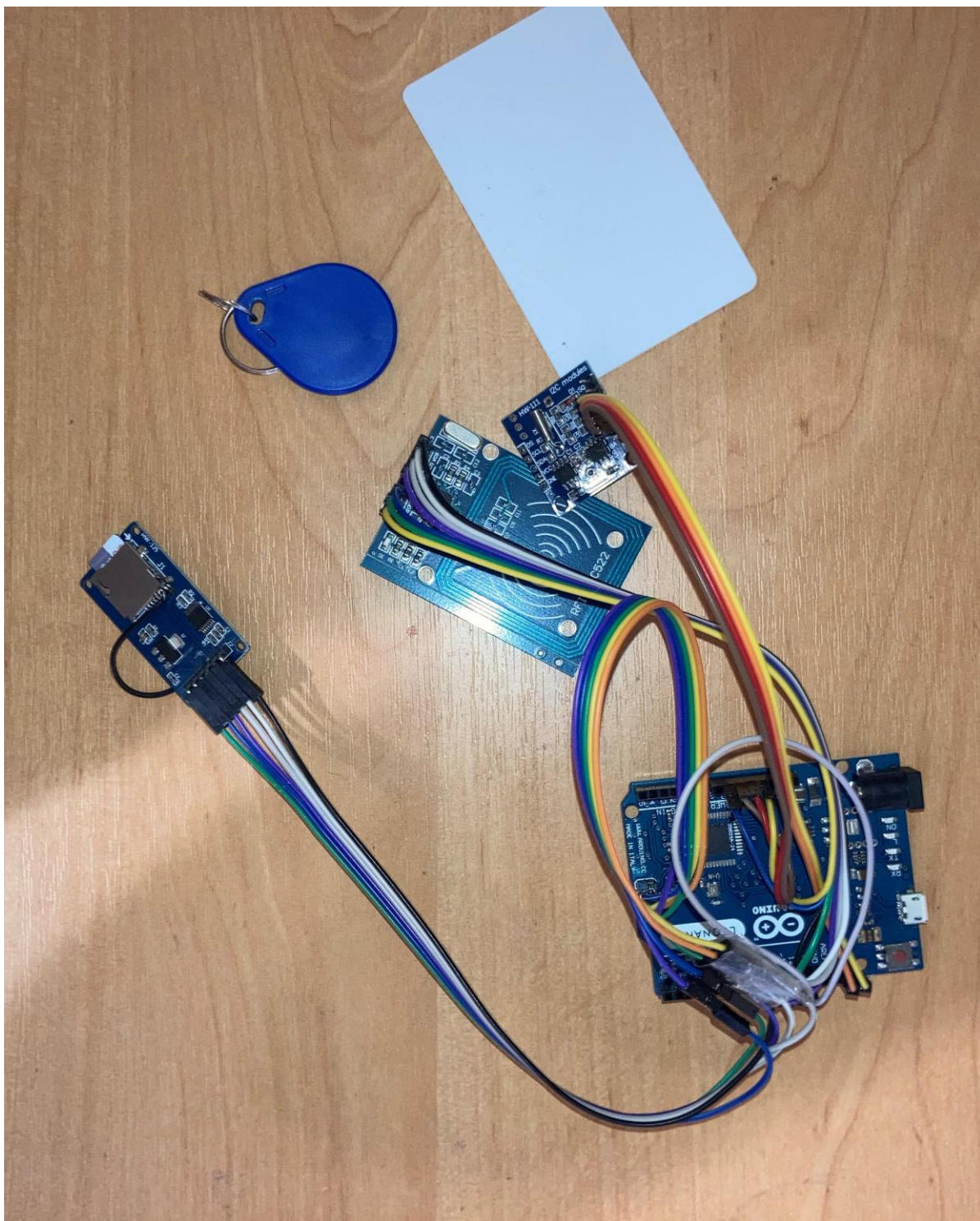
```

```
else date1 = String(tm.Day,DEC);
if (tm.Month<10){
date1= date1 + "_" + "0" + String(tm.Month, DEC);
}
else date1 = date1 + "_" + String(tm.Month,DEC);
date1 = date1 + "_" + tmYearToCalendar(tm.Year);
return date1;
}
#endif
```


СХЕМА РОЗРОБЛЕНОГО ПРИБАДУ



ДОДАТОК В
ЗОВНІШНІЙ ВИГЛЯД РОЗРОБЛЕНОГО ПРИБЛАДУ



ВІДГУК
керівника економічного розділу
на кваліфікаційну роботу бакалавра
на тему:
«Розробка апаратного логгеру для персональних комп'ютерів на базі
мікроконтролерів Arduino »
Студента групи 122-19ск-2 Носуля Олега Павловича

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Nosul_dipl.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Nusul_dipl.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF.
Програма	
Arduino.zip	Архів. Містить коди програми.
Презентація	
Nosul.pptx	Презентація кваліфікаційної роботи.