**Дніпровський ліцей інформаційних технологій**

**при Дніпровському національному університеті**

**імені Олеся Гончара**

**Випускна робота**

**на тему:**

**«Автоматизація обліку робочого часу»**

**Виконавець:**

**ліцеїст 11-Г класу**

**Устінов Іван**

**Науковий керівник:**

**Чашка Ю. М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дніпро**

**2020**

**Зміст**

[Вступ 3](#_Toc59636644)

[Актуальність: 3](#_Toc59636645)

[Метою 3](#_Toc59636646)

[Тема 3](#_Toc59636647)

[Основна частина 4](#_Toc59636648)

[Теоретична частина з робототехніки. 4](#_Toc59636649)

[Теоретична частина з програмування. 8](#_Toc59636650)

[Опис роботи 11](#_Toc59636651)

[Блок схема 14](#_Toc59636652)

[Напрямки використання 15](#_Toc59636653)

[Програмно-апаратні вимоги 15](#_Toc59636654)

[Комплектація програми 15](#_Toc59636655)

[Використані програмні засоби 15](#_Toc59636656)

[Висновки 16](#_Toc59636657)

# Вступ

Актуальність: Людям постійно не вистачає часу. Люди схильні відволікатись на безліч чинників, які заважають виконанню найважливіших завдань. В кінці робочого дня ми часто не розуміємо на що було витрачено робочий час та чому ми нічого не встигаємо. Таке, мабуть, бувало у кожної людини. Для вирішення цієї проблеми використовують різні додатки для планування часу, але всі вони працюють на телефонах або комп’ютерах, що призводить до відволікання при використанні. Отже:

Метою моєї роботи була розробка пристрою для обліку часу по кожному виду діяльності.

Тема: автоматизація обліку робочого часу.

# Основна частина

## Теоретична частина з робототехніки.

Основною метою роботи була розробка незалежного пристрою, який може працювати без комп’ютера. Пристрій складається з плати Nodemcu на базі мікроконтролера esp8266, Lcd 16\*2 дисплею та енкодера, який є основним елементом управління. Корпус було розроблено та надруковано на 3d принтері.

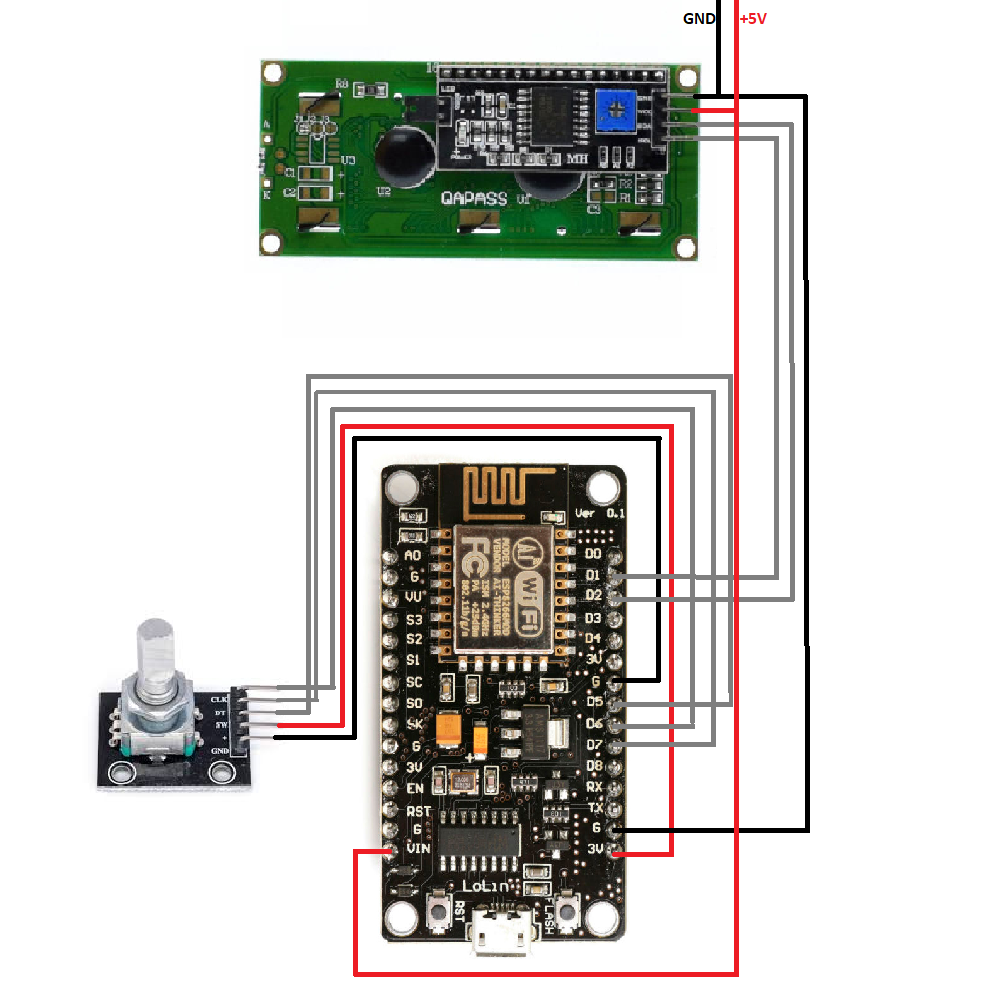


Рисунок 1 Схема пристрою

Рис. 1

Розглянемо дисплей. Він працює по шині i2c, яка дозволяє керувати ним через 2 контакти: SDA та SCl. Для роботи з шиною I2C використовуються дві бібліотеки: LiquidCrystal\_I2C.h та Wire.h.

Задачі, часу запланований час та статус кожної задачі зберігаються в енергонезалежній пам’яті EEPROM (Electrically Erasable Programmable Read-Only Memory). Збереження інформації відбувається після кожного її оновлення(отримання даних з комп’ютера, зміна затраченого або запланованого часу або очищенні даних)

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Дані зчитуються з енергонезалежної пам’яті при запуску програми. Це дозволяє продовжити роботу після перезавантаження пристроюю.

Изображение выглядит как текст

Автоматически созданное описание

Для очищення даних необхідно запустити пристрій з натиснутою кнопкою енкодера. Всі дані про час та статус замінюються на 0. Оскільки для збереження інформації використовуємо змінну типу int, то під кожну комірку даних виділяється 4 байти в постійній пам’яті, тому ми замінюємо елемент пам’ятіб що знаходиться по адресу І\*4 на змінну int a, яка дорівнює 0;

Изображение выглядит как текст

Автоматически созданное описание

Найскладнішим в програмуванні пристрою було розробити зручний для користувача інтерфейс, оскільки виведення даних відбувається на маленький дисплей та єдиним елементом управління є енкодер. Переходи між пунктами меню реалізовано так: при виборі на інший пункту меню всі змінні типу bool, що відповідають за кожний пункт меню приймають значення false, а та змінна, що відповідає за вибраний пункт меню приймає значення true.

Изображение выглядит как текст

Автоматически созданное описание

**Комунікація пристрою з комп’ютером.**

З’єднання пристрою з комп’ютером відбувається через послідовний порт (СОМ-порт). Пристрій через USB-порт комп’ютера.

* Відправка даних реалізована стандартними для плати Nodemcu командами *Serial.print()* та *Serial.println().*

Изображение выглядит как текст

Автоматически созданное описание

* Отримання даних відбувається коли функція *Serial.available()* повертає число більше нуля, тобто коли кількість доступних байт в буфері більше нуля. Дані з комп’ютера відправляються з затримкою 100мсб тому можливо зчитувати кожну строку окремо. Це дозволяє уникнути використання зайвого массива та розділення його на строки.

Изображение выглядит как текст

Автоматически созданное описание

## Теоретична частина з програмування.

Для написання програми використовувалась середа розробки Microsoft Visual Studio 2015. Мова програмування – C#. Були використані такі елементи Windows Forms:

* Button
* DataGridView
* Label
* Textbox
* Treeview
* Combo box
* SerialPort

Підключення до пристрою відбувається з-за допомогою елементу SerialPort. Розглянемо форму «Connect to device». При натисканні кнопки «Check ports» скануються всі доступні COM-порти та виводяться в ComboBox. Якщо не знайдено жодного доступного порту програма видасть попередження.

Изображение выглядит как текст

Автоматически созданное описание

Далі, для підключення до пристрою необхідно обрати порт, до якого він підключений, та натиснути кнопку Connect. Встановлена найбільша доступна символьна швидкість – 2000000 Бод, оскільки вся інформація відправляється однією строкою і при меншій швидкості деякі дані могли

Изображение выглядит как текст

Автоматически созданное описаниебути втрачені.

Далі дані, отримані від пристрою, записуються в строку «recieveddata». Дані відправляються у виді таблиці і розділені пробілом. Кожна задача відділена від попередньої символом нової строки. Отже програма отримує строку у такому вигляді:

|  |  |  |  |
| --- | --- | --- | --- |
| Задача 1 | Час 1 | Запланований час 1 | Статус 1 |
| Задача 2 | Час 2 | Запланований час 2 | Статус 2 |
| Задача 3 | Час 3 | Запланований час 3 | Статус 3 |

Изображение выглядит как текст

Автоматически созданное описаниеСпочатку розбиваємо цю строку за знаком нового рядка (/n) у масив «string1». Після цього кожен елемент масиву «string1» розбивається за знаком пробіл у двомірний масив «data». Далі масив «data» виводиться у DataGridView. Усі дані зберігаються у секундах, тому спочатку час необхідно перевести у години та хвилини.

Відправка задач на пристрій виконується майже так само як і з пристрою, але відправляються лише назви задач оскільки запланований час та потрачений час можна змінити на пристрої.

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание

## Керівництво для користувача

Завантаживши програму ви побачите титульну форму, з якої можна перейти до теорії або до форми, з якої можна перевірити час по кожній задачі:

У розділі «Теорія» знаходиться основна інформація про засоби збереження часу та інструкція використання програми.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рис. 2

У розділі «Хронометраж по видам діяльності» знаходиться основна сторінка програми. У даному розділі можна з’єднатися з пристроєм та отримати всю інформацію з пристрою про кожну задачу. Для цього необхідно натиснути кнопку «Check ports» для знаходження доступних послідовних портів. Далі, обравши порт, до якого підключено пристрій, необхідно натиснути «Connect» для з’єднання.

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3

Далі можна заповнити стовбець «Task» (**Але важливо**: необхідно заповнити всю таблицю, оскільки в іншому випадку на пристрої будуть пусті задачі). Після заповнення усього стовбця необхідно натиснути кнопку «Send data» і список задач на пристрої зміниться.

Для отримання даних з пристрою необхідно натиснути кнопку «Send» на ньому. У DatagridView відобразиться вся інформація про кожну задачу. Задачі можна сортувати по статусу (закінчені та незакінчені).

Изображение выглядит как стол

Автоматически созданное описание

Рис. 4

## Блок схема

Необроблені

дані

Оброблені

дані

Блок схема 1

## Напрямки використання

Кожна людина зустрічалась з проблемою нестачі часу. Існує багато програм для планування часу, але всі вони знаходяться на комп’ютері або телефоні, що призводить до відволікання при використанні. Оскільки розроблений мною пристрій працює незалежно від комп’ютера, то його можна використовувати для моніторингу часу будь-якої задачі. Наприклад при виконанні домашніх справ (домашнього завдання, ігри хоббі та інше) або на будь-якій фірмі для контролю персоналу.

## Програмно-апаратні вимоги

* ОС Windows 7/8/8.1/10 з архітектурою 32 розряди(x86) або 64 розряди (х64).
* Процесор з частотою 1.6 ГГц або більш потужний.
* ОЗУ об'ємом 1 ГБ.
* 50 МБ доступного простору на жорсткому диску.
* Жорсткий диск з частотою обертання 5400 об / хв.
* Роздільна здатність екрану не менше 1200x780.
* Заінстальовані: середа розробки Visual Studio 2015

## Комплектація програми

* Exe-файл – vipusk.exe
* Файли з розширенням .txt – текст
* Папка Program – папка з програмою для пристрою

## Використані програмні засоби

* Arduino Ide
* Visual Studio 2015

# Висновки

Під час виконання цієї роботи, мною було оброблено безліч матеріалу. Були досліджені особливості програмування плати NodeMcu та особливості підключення інших елементів пристрою до неї.

Результатом роботи є готовий пристрій, який допоможе організувати свій робочий час. Пристрій може працювати без підключення до комп’ютеру, Дані з пристрою можна переглядати на комп’ютері, що є дуже зручним для перегляду великої кількості задач.