МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. І. СІКОРСЬКОГО»

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ

КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ АПАРАТУРИ

Звіт з практичної роботи у середовищі LabVIEW

з дисципліни:«Вимірювальні перетворювачі фізичних величин»

на тему:“Дослідження характеристик Li-ion акумулятора”

Виконали:

студенти гр. ДК-51

Сорокін Д.А.

Тимошенко С.В.

Перевірив:

доц. Яганов П.О.

Київ – 2018

**Зміст**

1. Актуальність дослідження
2. Об’єкт дослідження
3. Мета дослідження
4. Теоретичні відомості
5. Хід роботи
6. Процес отримання даних
7. Побудова метрологічної характеристики
8. Висновки
   * + 1. **Актуальність дослідження**

В наш час людство використовує дуже багато девайсів, які працюють на автономному живленні. Кожен день ви використовуєте пристрій, що містить у собі акумулятор. Нажаль поки не існує ідеального джерела живлення, та акумулятори з часом розряджаються та іх необхідно під заряджати. Цей процес не завше є простим, через певні фізичні властивості акумуляторів. Тому важливим є дослідження характеру поведінки та принципу роботи таких популярних приладів, як акумулятор.

* + - 1. **Об’єкт дослідження**

В даній роботі будуть досліджуватися метрологічні характеристики літій-іонного акумулятора. В якості досліджуваного зразка був обраний звичайний акумулятор ємністю на 280 мАч та вихідною напругою 3,7 В.

* + - 1. **Мета дослідження**

Для дослідження, зазвичай, необхідно мати спеціальне обладнання, що дозволить знімати характеристики то контролювати процес. Не кожний має у розпорядженні осцилограф, чи мультиметр, але під рукою може бути ноутбук. За допомогою нього та програмного забезпечення LabVIEW можна з легкістю побачити ступінь заряду акумулятора. Більш того, за допомогою середовища LabVIEW можна вимірювати та досліджувати різні параметри, в залежності від потреби користувача.

Дана робота дозволяє дослідити деякі параметри і передати їх прямо на екран комп’ютера, що є дуже зручним для користування. Так у даній роботі буде розглянуто:

1. Дослідження принципів перетворення фізичної величини в електричний сигнал.
2. Створення вимірювального приладу за допомогою вимірювального перетворювача фізичної величини, платформи Arduino та середовища LabVIEW.
3. Побудова метрологічної характеристики заряду/розряду літій-іонного акумулятора.
4. **Теоретичні відомості**

Звичайний розряд - це живлення електричної схеми (моделі).

Критичний розряд допускається до не менше, ніж 3.3B на банку. На короткий час можна розряджати до 3.0B, після чого бажано відразу (протягом години) поставити на зарядку, за умови, що батарейка не гаряча і не дуже замерзла. При 2.9В починаються незворотні зміни в електроліті, ємність буде знижуватися швидше. При розряді до 2.8В з електроліту почне виділятися сіль літію і осідати на електродах, що відразу знизить ємність (за рахунок зниження щільності електроліту), і струмовіддачі (за рахунок зменшення площі електродів). Також зменшується можлива кількість циклів. Крім того, акумулятор може і не зарядитися.

За іншими даними, при низькій напрузі (менше 2.5В) можлива металізація літію, створення струмопровідних літієвих «містків». І при зарядці такого акумулятора утворення провідних ділянок з літію призводить до внутрішнього короткого замикання та, як наслідок, до теплового розгону акумулятора з можливим вибухом акумулятора.

Для своєчасного попередження перезаряджання використовують звукові індикатори, які видають різного роду сигнали при підході напруги акумулятора до мінімальних значень. Краще використовувати пристрої, що дозволяють контролювати напругу побаночно.

Рекомендована мінімальна напруга, до якої можна доводити батарею при експлуатації, це 3.6В на банку. При цьому кількість циклів розряду / заряду без помітної втрати ємності може обчислюватися сотнями.

При розряді струмом I = C можливо близько 500 повноцінних циклів заряду-розряду.

При розряді струмом, багаторазово перевищує номінал ємності (звичайний режим роботи) кількість повноцінних циклів знижується в рази.

Найнебезпечніше для LiPo висока температура, а не низька напруга. Не слід при розряді допускати нагрів акумулятора більше, ніж 60 ° С.

При використанні на повітрі в холодний сезон акумулятори краще тримати в теплі перед використанням і після.

1. **Хід роботи**

Для початку акумулятор заряджається через спеціальну зарядну плату. Далі підключається до попередньо зібраної схеми струмового навантаження. За допомогою потенціометра, який знаходиться на схеми, ми можемо регулювати струм споживання, іншими словами, як швидко акумулятор буде розряджатись. Після цього підключаємо виводи до плати Arduino LEONARDO, щоб зчитати дані про заряд/розряд акумулятора, і саму Arduino підключаємо до комп’ютера. Схема підключення стенду зображена на Рис. 3.

|  |
| --- |
|  |
| Рис. 3. Стенд для вимірювання. |

Вимірювальний стенд складається з трьох складових:

1. Струмове навантаження.
2. Літій-іонний акумулятор.
3. Arduino.

Оскільки процес розряду акумулятора є доволі довгим процесом, для збільшення швидкодії нами було зібрано струмове навантаження, головною особливістю якого є імітація роботи реального приладу, що споживає постійний струм. Схему такого струмового навантаженнязображено на Рис. 4.

|  |
| --- |
|  |
| Рис. 4. Струмове навантаження. |

Принцип роботи оснований на базі операційного підсилювача LM358. Напруга живлення складає 12 вольт. Через подільник напруги на не інвертуючий вхід операційного підсилювача приходить 0,6 В. Коли на резисторі R1(інвертуючий вхід) формується напруга більша, ніж напруга на неінвертуючому вході, то вихід ОП встановлюється в 0, транзистор буде закритим і напруга на резисторі R4 зменшиться. Тоді напруга на прямому вході буде більша, що змусить вихід ОП перейти в стан 1. Це відкриє транзистор, і в наслідок цього, напруга на R4 збільшиться.

Спостерігаємо циклічний процес автобалансування схеми, що дозволяє отримувати на резисторі R4 постійну напругу(як наслідок постійний струм), незалежно від напруги акумулятора. Значення цього струму можна варіювати зміною напруги на неінвертуючому вході, тобто зміною опору резистору R1.

В результаті поєднання навантаження та акумулятор, другий почне розряджатися. Напруга заряду акумулятора надходить на Arduino, проходить процес оцифровуваннята передається до комп'ютера.

1. **Отримання даних**

Діапазон напруг акумулятора складає від 0 до 3.7 В. Розрядність АЦП (ADC) Arduino 10 біт, що означає, що є 1024 можливих станів (одиниці АЦП).

Напруга, яку сприймає Arduino, на аналоговому порту становить 5В.

Тому:

1024ADCunits / 5V = 204,8 ADCunits / V(1)

Використовуючи дане співвідношення, ми зможемо перетворити одиниці АЦП Arduino в напругу.

Для того, щоб передати значення через Arduino, необхідно завантажити в неї наступний код:

constintanalogInPin = A0;

constintanalogOutPin = 9;

intsensorValue = 0;

intoutputValue = 0;

voidsetup() {

Serial.begin(9600);

while(!Serial);

}

voidloop() {

sensorValue = analogRead(analogInPin);

outputValue = map(sensorValue, 0, 1023, 0, 255);

analogWrite(analogOutPin, outputValue);

delay(200);

}

В результаті коректного підключення, данні нашої напруги будуть надходити до ноутбука. Залишилось лише ці данні прочитати.

1. **Побудова метрологічної характеристики**

**Програма**

Для коректного корегування даних та їх відображення у середовищі LabVIEW необхідно написати програму. В нашому випадку результат роботи зображено на Рис.1

|  |
| --- |
|  |
| Рис.1 Вікно коду. |

На Рис. 2 зображено лицеву панель приладу.

|  |
| --- |
|  |
| Рис.2. Лицева панель приладу. |

**2. Процес створення коду**

Дану роботу можна розбити на 5 логічних блоків:

* Читання даних з Arduino Leonardo
* Підрахунок середнього арифметичного
* Зберігання/читання даних у файл
* Вивід даних на екран
* Блок корегування напруги у відсотки
  1. **Читання даних з Arduino**

Для виконання даного блоку нам знадобляться наступні компоненти. Шлях додання яких зображено нижче.

|  |
| --- |
| Screenshot_2.jpg |
| Screenshot_4.jpg |
| Рис. 3. Додання елементів на блок діаграму |

Таблиця 1. Детальний опису компонентів

|  |  |
| --- | --- |
| Опис | Використання |
| Screenshot_5.jpg | Для читання даних з COM-порту |
| Screenshot_3.jpg | Для налаштування роботи з COM-портом |
| Screenshot_6.jpg | Для закриття COM-порта |

На екрані користувача додаємо наступні складові:

|  |
| --- |
| Screenshot_1.jpg |
| Рис.4. Додання елементів на панель користувача. |

Таблиця 2. Детальний опису компонентів

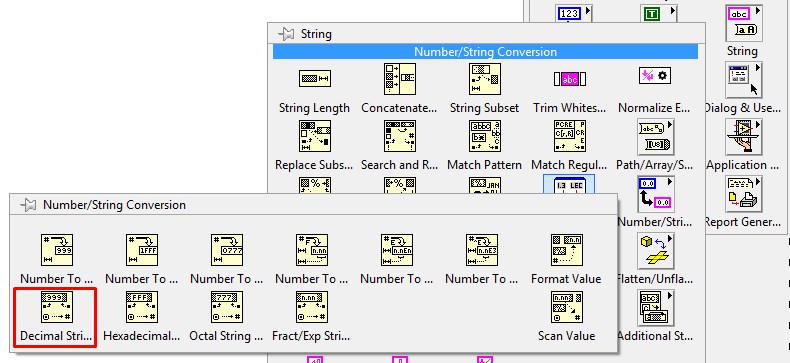
|  |  |
| --- | --- |
| Назва | Опис |
|  | Обрання COM-порта, до якого підключено Arduino |
| Screenshot_7.jpg | Визначення швидкості передання даних |

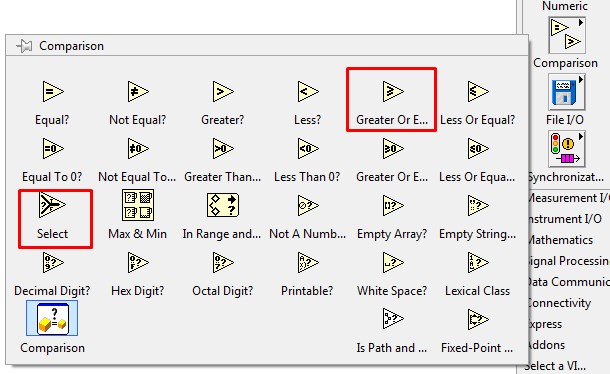
|  |
| --- |
|  |
| Screenshot_9.jpg |
| Рис 5. Формат підключення елементів. |

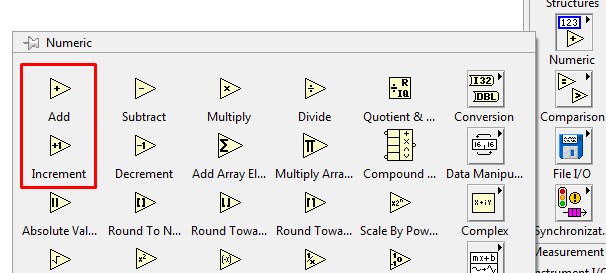
В результаті ми маємо інтерактивний блок коду, в якому можна вибрати з яким COM-портом ми будемо працювати. З нього будуть надходити дані на певній швидкості, яку ми також можемо задати. І в цьому ж блоці можемо прочитати дані для майбутньої роботи з ними.

* 1. **Підрахунок середнього арифметичного**

Для виконання даного блоку вам знадобляться наступні компоненти. Шлях додання яких зображено нижче.







|  |  |
| --- | --- |
| Опис | Використання |
| Screenshot_17.jpg | Конвертація данних з Arduino з формату String у Integer. |
| Screenshot_14.jpg | Операція «більше-дорівнює». Порівнюємо кількість циклів ітерацій зі встановленою константою (визначаємо для якої кількості елементів будемо рахувати середнє значення). |
| Screenshot_16.jpg | Операція вибору. Залежно від того, чи виконалися N циклів, обираємо які данні будуть надходити далі. Працює аналогічно оператору If в С++ |
| Screenshot_15.jpg | Операція інкрементації. Необхідна для підрахунку кількості тактів, які запустив цикл. |
| Screenshot_13.jpg | Операція складання. Додає до вже існуючої суми нове значення, що надійшло. |

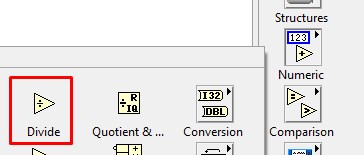
|  |
| --- |
| Screenshot_18.jpg |
| Рис. 6. Формат підключення елементів. |

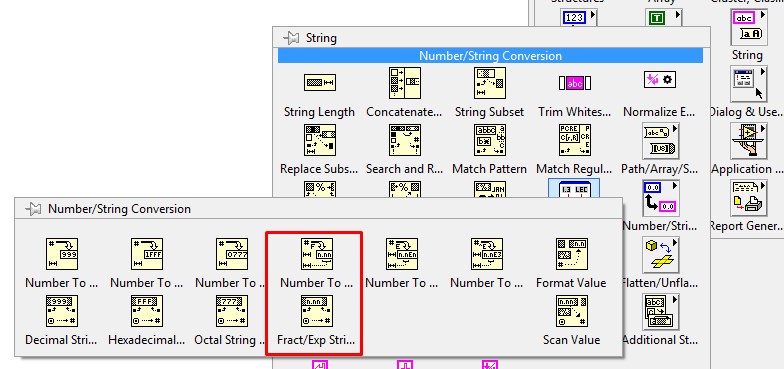
Під кінець виконання даного блоку програми отримали суму з 50-ти значень, що надійшли з Arduino та по яким буде розраховано середнє значення. Далі вони пройдуть до блоку, який конвертує ці дані для зберігання їх у файл.

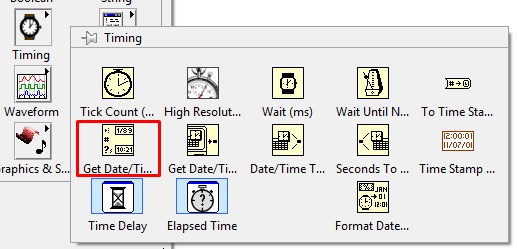
* 1. **Зберігання/читання даних у файл**

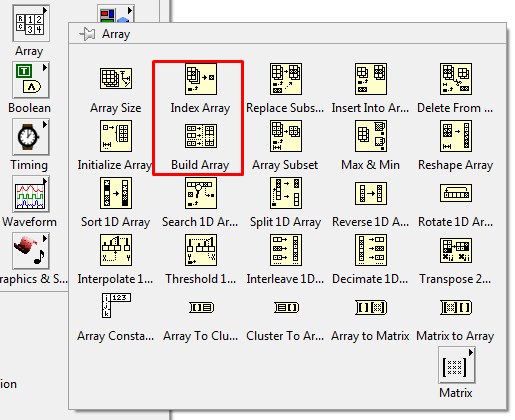
Для виконання даного блоку вам знадобляться наступні компоненти. Шлях додання яких зображено нижче.











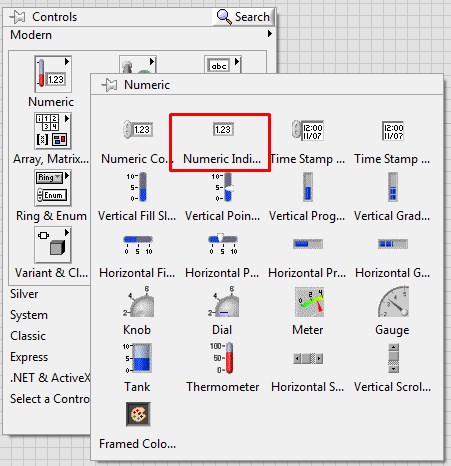
|  |  |
| --- | --- |
| Опис | Використання |
| Screenshot_24.jpg | Операція ділення. Ділимо суму, отриману на попередньому кроці, на 50, щоб отримати середнє значення. |
| Screenshot_25.jpg | Конвертуємо числовий формат середнього значення. |
| Screenshot_26.jpg | Отримаємо дату та час для побудови часових характеристик. |
| Screenshot_27.jpg | Створюємо масив із значень, що надходять. |
| Screenshot_28.jpg | Конвертуємо значення. |
| Screenshot_29.jpg | Додаємо нові елементи до вже існуючого масиву даних. |
| Screenshot_30.jpg | Відкриваємо файл (таблицю) для читання. |
| Screenshot_31.jpg | Відкриваємо файл (таблицю) для запису нового значення. |

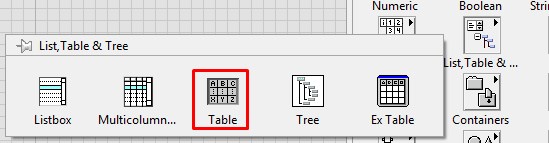
|  |
| --- |
|  |
| Рис 7. Формат підключення елементів. |

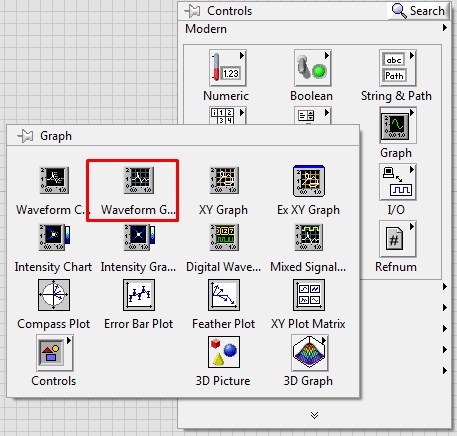
Результатом роботи став блок, в якому розраховується середнє значення даних. Це потрібно через те, що дані з Arduino приходять з певною неточністю і тому, задля уникнення небажаних і некоректних результатів був організований підрахунок середнього значення. Також в цьому блоці було організовано створення файлу, в який надходять і записуються значення.

* 1. **Вивід значень на екран**

Для виконання даного блоку вам знадобляться наступні компоненти. Шлях додання яких зображено нижче.







|  |  |
| --- | --- |
| Назва | Опис |
| Screenshot_36.jpg | Таблиця в якій будуть відображатися дані. |
| Screenshot_38.jpg | Граф, на якому буде будуватися характеристика значень, що відображатимуться у таблиці. |
| Screenshot_37.jpg | Числовий індикатор, що відтворюватиме значення величини. |

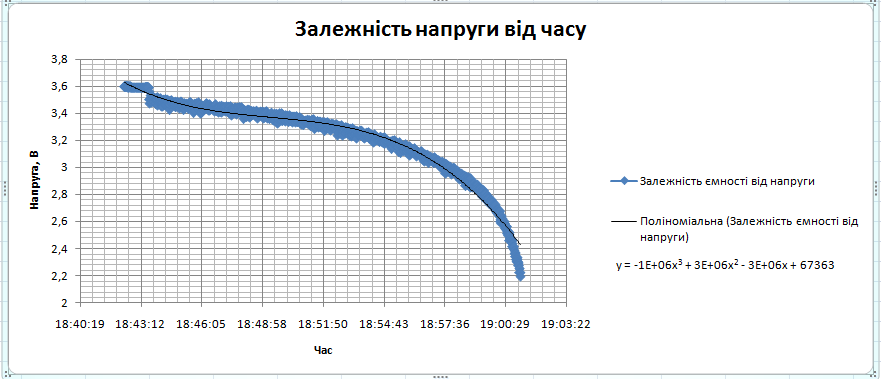
|  |
| --- |
|  |
| Рис 8. Формат підключення елементів. |

До вже існуючого необхідно додати індикатори, на яких будуть відображатись значення, так, як зображено на рисунку.

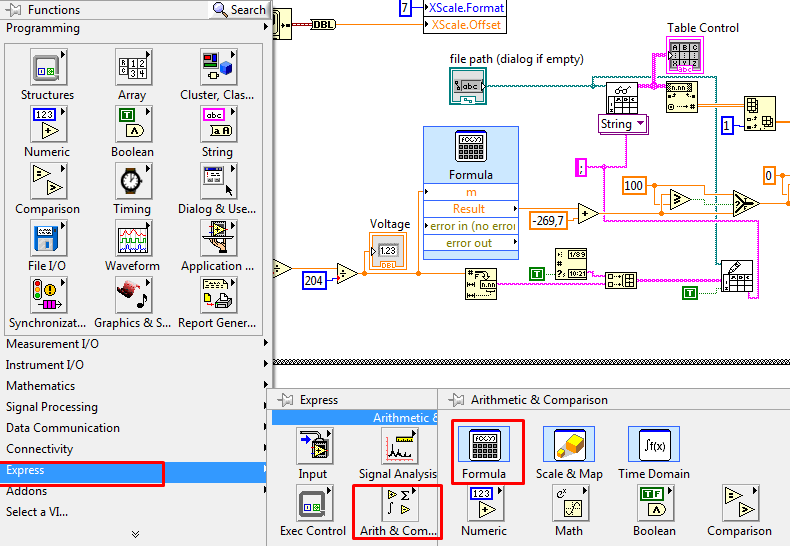
Результатом виконання цього блоку було отримано вікно користувача, в якому значення відображаються, як і в певних “комірках”, так і на графіку залежності.

**2.5 Блок корегування**

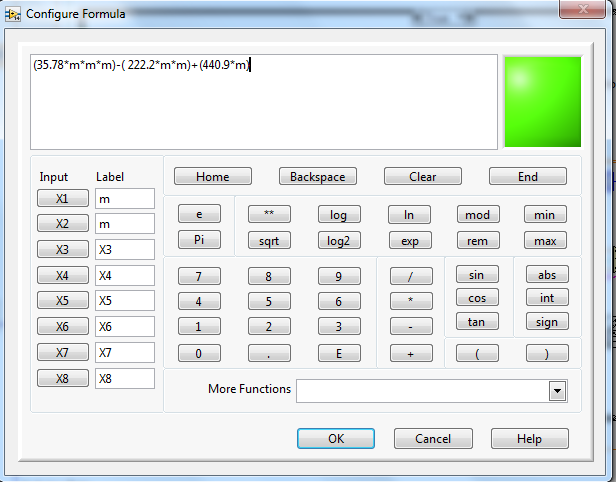
Записані дані були експортовані в Excel, по яким був побудований графік залежності вихідної напруги від часу. Отриманий графік був конвертований в залежність ємності від напруги, а після цього - апроксимований поліноміальною лінією третього порядку, так як вона найточніше описала поведінку графіка. Вивели на екран рівняння лінії тренду, яке в майбутньому використаємо для корегування відображення ємності в відсотках.



Щоб зробити це програмно нам потрібен наступний елемент. Його можна знайти за таким шляхом: ПКМ-Express-Arithmetic&Comparison-Formula.



Додавши елемент на схему і клікнувши на нього два рази, відкриється наступне вікно:



Тут ми можемо задати формулу за якою будуть розраховуватись вихідні значення залежно від вхідних. Сюди ж ми пишемо рівняння нашої апроксимуючої лінії.

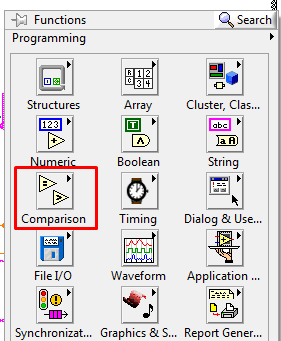
В результаті роботи даної програми було виявлено, що в деяких випадках значення розряду в відсотках відображалось некоректно. Це пов’язано з тим, що точність вимірів не ідеальна, а апроксимація має певну похибку. Значення могли бути менше 0% або більше 100%. Щоб уникнути цієї неприємності був створений невеликий блок програми, який аналізує значення на виході і корегує їх при необхідності.

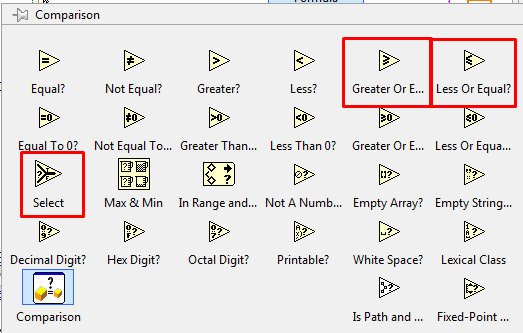
Вигляд корегуючого блоку зображений на Рис. 9.



Рис. 9. Корегуючий блок.

Щоб його створити нам знадобляться елементи, які знаходять у вкладці Comparison.





Робота блоку основана на тому, що значення спочатку зрівнюються з числом 100. Якщо воно більше 100, то на екрані буде видно 100%, в інакшому випадку дані будуть передані на наступний елемент блоку, який порівняє це число з 0. Якщо воно більше 0, на екрані побачимо число, яке пройшло всі перевірки, в іншому випадку, якщо воно менше 0, то на екрані побачимо 0%.

**Висновок:**

В даній роботі було проведено дослідження розряду акумулятора. Було створено вимірювальний стенд для отримання та читання характеристики розряду. Дані, що надходили у середовище LabVIEW, конвертувалися у зрозумілий формат виводилися на графіку, та зберігалися у файл для подальшого аналізу.

Середовище LabVIEW є досить універсальним та дозволяє досліджувати різноманітні процеси без використання спеціального обладнання, що є досить зручним у наш час.