

Розробка заняття гуртка
«Радіоелектронного конструювання та основ
робототехніки»

***Тема: «Логічні операції та вирази. Оператор
розгалуження».***

Керівник гуртка: Михайленко В. В.

Тема. Логічні операції та вирази. Оператор розгалуження.

Мета:

- ✓ **навчальна:** навчити вихованців пояснювати правила виконання операцій кон'юнкції, диз'юнкції та заперечення над величинами логічного типу; навчити складати алгоритмічні конструкції на основі оператора розгалуження.
- ✓ **розвивальна:** розвивати вміння правильно і чітко висловлювати свої думки, грамотно формулювати твердження;
- ✓ **виховна:** виховувати інформаційну культуру, обайливе ставлення до комп'ютерної техніки та навчального обладнання.

Обладнання та наочність: інтерактивна дошка (екран), комп'ютери, навчальна презентація, навчальний набір «IArduino».

Програмне забезпечення: середовище програмування «Arduino IDE».

Хід заняття

I. Організаційний етап

- привітання
- перевірка присутніх
- перевірка готовності вихованців до заняття

II. Мотивація навчальної діяльності

Вступна бесіда

Мабуть ви не одноразово спостерігали що певні події відбуваються за певних умов. Наприклад, натискаємо вимикач – вмикається світло, підставляємо руки під кран – тече вода, підносимо смартфон до вуха при розмові – вимикається його дисплей. Тобто існують події в залежності від яких, виконуються певні дії. Отже, на сьогоднішньому занятті ми, за допомогою мови програмування, навчимося описувати та опрацьовувати події.

III. Теоретична частина

1. Логічні операції та вирази.

Логічним виразом називають вираз, який може набувати одне з двох значень: **true** або **false**. Висловлювання є прикладом логічного виразу. Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює true. Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює false. Змінні, які можуть набувати одне з двох значень true

або false, називають змінними логічного типу. Логічний тип змінних позначається boolean.

Змінні логічного типу також вважаються логічними виразами.

<i>x</i>	<i>not x</i>
true	false
false	true

Таблиця істинності для заперечення

<i>x</i>	<i>y</i>	<i>x and y</i>
true	true	true
true	false	false
false	true	false
false	false	false

Таблиця істинності для кон'юнкції

<i>x</i>	<i>y</i>	<i>x or y</i>
true	true	true
true	false	true
false	true	true
false	false	false

Таблиця істинності для диз'юнкції

У логічних виразах для логічних операцій, розглянутих вище, визначено такий пріоритет виконання операцій: спочатку виконуються операції заперечення, потім – операції кон'юнкції, потім – операції диз'юнкції. Для змінення цього порядку виконання логічних операцій використовують дужки. Для обчислення значень логічного виразу можна використати таблицю істинності. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу.

2. Оператор розгалуження IF ELSE

Оператор розгалуження (умовний оператор) IF ELSE – алгоритмічна структура, що дозволяє виконати деяку команду або серію команд в залежності від значення логічного виразу. Оператор розгалуження належить до операторів керування. Оператори керування визначають, у якій послідовності виконується програма. Якби їх не було, оператори програми завжди виконувалися б послідовно, у тім порядку, у якому вони записані.

Оператори розгалуження (умовні оператори) дозволяють вибрати один з варіантів виконання дій залежно від яких-небудь умов. Умова – це логічне вираз, тобто вираз, результатом якого є логічне значення true (істина) або false (неправда).

Оператор if вибирає один із двох варіантів послідовності обчислень.

Загальний вигляд оператора IF ELSE:

Повне розгалуження

```
If (condition)           // якщо умова condition істинна тоді
  {Block1}               // спрацьовує блок 1 операторів
else                     // якщо умова condition не істинна тоді
  {Block2}               // спрацьовує блок 2 операторів
```

Неповне розгалуження

```
If (condition)           // якщо умова condition істинна тоді
  {Block1}               // спрацьовує блок 1 операторів
```

Можна повне розгалуження замінити двома неповними, наприклад повне розгалуження

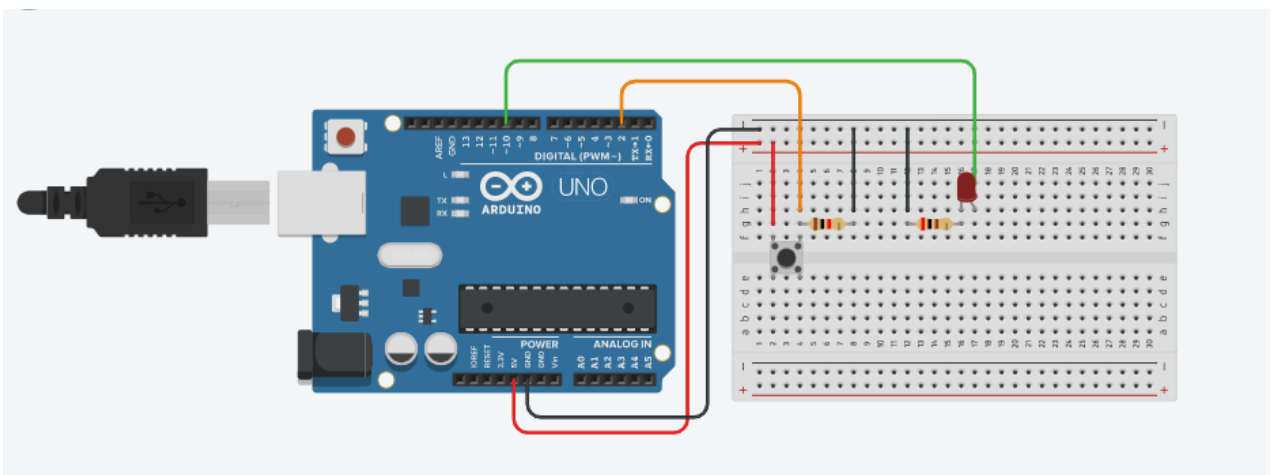
```
If (condition)           // якщо умова condition істинна тоді
  {Block1}               // спрацьовує блок 1 операторів
else                     // якщо умова condition не істинна тоді
  {Block2}               // спрацьовує блок 2 операторів
```

можна замінити двома неповними

```
If (condition)           // якщо умова condition істинна тоді
  {Block1}               // спрацьовує блок 1 операторів
If (!condition)          // якщо умова condition не істинна тоді
  {Block2}               // спрацьовує блок 2 операторів
```

V. Практична частина

1. Складіть пристрій за схемою



2. Завантажте код програми в плату Arduino

```
/* Створюємо константу типу int для збереження вивода
до якого під'єднана кнопка */
const int PinKey = 2;
/* Створюємо константу типу int для збереження вивода
до якого під'єднаний світлодіод */
const int PinLed = 10;
/* Створюємо змінну типу bool для збереження стану
кнопки (1 - натиснута, 0 - не натиснута)*/
bool Action = 0;
void setup()
{
  /* Задаємо вивід PinKey як вхід*/
  pinMode(PinKey, INPUT);
  /* Задаємо вивід PinLed як вихід*/
  pinMode(PinLed, OUTPUT);
}
void loop()
{
  Action = digitalRead(PinKey);           // Зчитуємо значення з PinKey
  if (Action == 1)                       // Якщо Action = 1
  {digitalWrite (PinLed, HIGH);}         // Вмикаємо світлодіод
  else digitalWrite (PinLed, LOW);      // Інакше світлодіод вимкнено
}
```

3. Перевірте роботу пристрою (*при натисканні кнопки світлодіод вмикається при відпусканні вимикається*)

4. Замініть в кодї програми повне розгалуження двома неповними.

V. Підсумки заняття

На занятті я навчився ...

На занятті я зрозумів...

На занятті мені сподобалось...