

## **Визуальный редактор для построения программ с генерацией кода на встроенном языке.**

### **Функциональные требования.**

1. Редактирование программы в графическом виде, где каждый оператор и переменная представлены блоком картинкой с некоторыми полями ввода. В качестве примера реализации можно посмотреть Scratch(<http://scratched.media.mit.edu>) или LabView(<http://www.ni.com/labview>).
2. Переменные в блоках должны отображаться не по имени из встроенного языка, а в виде текстового описания.
3. Если переменная может принимать ограниченное число значений, предусмотреть в блоке выбор из текстовых описаний, а не выбор числового значения.
4. Генерация кода по графическому представлению.
5. Построение графического представления по тексту программы.

### **Требования к реализации.**

1. Язык программирования C++.
2. GUI библиотека Qt 4.7.4.
3. Инструментарий для сборки Microsoft Visual C++ Compiler 10.0 (x86).
4. Редактор должен быть реализован как наследник от QWidget.

## Описание встроенного языка.

- язык имеет C-подобный синтаксис;
- вся программа записывается одной строкой без пробелов, табуляций и переводов каретки;
- имена переменных и операторов регистрозависимые.

## Синтаксис программы.

РБНФ:

Синтаксис = "IF(" Выражение "){" Группа операторов присвоения "} [ "ELSE{" Группа операторов присвоения "} ]".  
Выражение = "(" Операнд ")" Оператор "(" Операнд ")".  
Операнд = Переменная | Выражение | Число.  
Оператор = "&&" | "|" | ">=" | "<=" | "==" | "!=" | "&" | "|".  
Переменная = "IN0" | "IN1" | "IN2" | "IN3" | "DisableIN0" | "DisableIN1" | "DisableIN2" | "DisableIN3" | "OUT0" | "OUT1" | "OUT2".  
Число = Цифра{Цифра }.  
Цифра = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".  
Группа операторов присвоения = Оператор присвоения { Оператор присвоения }.  
Оператор присвоения = Переменная "=" Число ";".

При истинности Выражения выполняется первая группа операторов присвоения, иначе, при наличии ELSE, вторая.

### Выражение.

Выражение состоит из 2 операндов и оператора, результатом выражения является целое число, где 0 – ложь, любое другое значение – истина. Каждый операнд заключён в скобки.

### Оператор.

Оператор	Описание															
&&	Логическое И															
	<table border="1"><thead><tr><th>ОП1</th><th>ОП2</th><th>Возвращаемое значение</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>Не 0</td><td>0</td></tr><tr><td>Не 0</td><td>0</td><td>0</td></tr><tr><td>Не 0</td><td>Не 0</td><td>1</td></tr></tbody></table>	ОП1	ОП2	Возвращаемое значение	0	0	0	0	Не 0	0	Не 0	0	0	Не 0	Не 0	1
	ОП1	ОП2	Возвращаемое значение													
	0	0	0													
	0	Не 0	0													
Не 0	0	0														
Не 0	Не 0	1														
	Логическое ИЛИ															
	<table border="1"><thead><tr><th>ОП1</th><th>ОП2</th><th>Возвращаемое значение</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>Не 0</td><td>1</td></tr><tr><td>Не 0</td><td>0</td><td>1</td></tr><tr><td>Не 0</td><td>Не 0</td><td>1</td></tr></tbody></table>	ОП1	ОП2	Возвращаемое значение	0	0	0	0	Не 0	1	Не 0	0	1	Не 0	Не 0	1
	ОП1	ОП2	Возвращаемое значение													
	0	0	0													
	0	Не 0	1													
Не 0	0	1														
Не 0	Не 0	1														
==	Возвращает 1, если операнды равны, иначе 0.															
!=	Возвращает 1, если операнды не равны, иначе 0.															
>=	Возвращает 1, если ОП1 больше либо равен ОП2, иначе 0.															
<=	Возвращает 1, если ОП1 меньше либо равен ОП2, иначе 0.															
>	Возвращает 1, если ОП1 больше ОП2, иначе 0.															
<	Возвращает 1, если ОП1 меньше ОП2, иначе 0.															
&	Битовое И. Возвращает число, полученное после применения оператора И для каждого бита ОП1 и ОП2 соответственно.															
	Битовое ИЛИ. Возвращает число, полученное после применения оператора ИЛИ для каждого бита ОП1 и ОП2 соответственно.															

### *Переменные.*

Все переменные представляют собой целые беззнаковые числа разрядностью 32 бита.

<b>Имя переменной</b>	<b>Описание</b>	<b>Доступна для чтения</b>	<b>Доступна для записи</b>
IN0	Значение на входе 0	+	
IN1	Значение на входе 1	+	
IN2	Значение на входе 2	+	
IN3	Значение на входе 3	+	
DisableIN0	Отключение подсчёта импульсов на входе 0. Если равно 1 – импульсы не считаются, 0 – считаются.	+	+
DisableIN1	Отключение подсчёта импульсов на входе 1. Если равно 1 – импульсы не считаются, 0 – считаются.	+	+
DisableIN2	Отключение подсчёта импульсов на входе 2. Если равно 1 – импульсы не считаются, 0 – считаются.	+	+
DisableIN3	Отключение подсчёта импульсов на входе 3. Если равно 1 – импульсы не считаются, 0 – считаются.	+	+
OUT0	Состояние выхода 0. 1- закрыт, 0-открыт.	+	+
OUT1	Состояние выхода 1. 1- закрыт, 0-открыт.	+	+
OUT2	Состояние выхода 2. 1- закрыт, 0-открыт.	+	+

### *Пример программы.*

```
IF((((IN0)>(400))&&((IN1)<(20))) | ((IN3)<(2))){OUT1=0;OUT2=1;OUT3=0;}
```