# **Лабораторная работа № 3**

Реализовать устройство, генерирующее сигнал заданной формы и отправляющее сгенерированные значения на компьютер посредством интерфейса USART.

**Использование прерываний обязательно**.

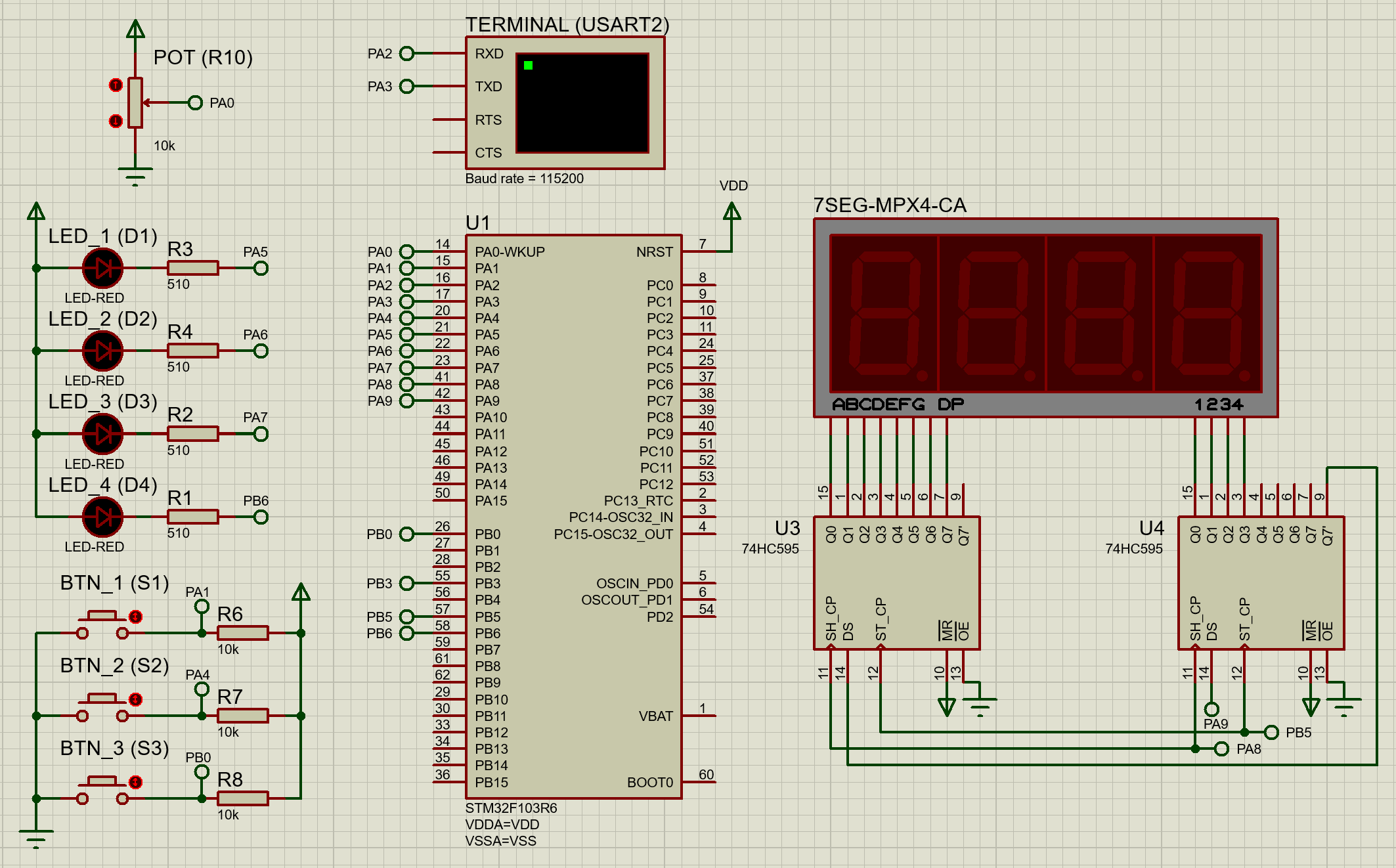
Описание платы расширения Arduino Multi-function Shield:

<http://publicatorbar.ru/2017/12/21/arduino-multi-function-shield/>

1. Осуществить конфигурацию контроллера STM32F103RBT6 (NUCLEO-F103RB) с помощью программы STM32CubeMX.
2. Написать код в среде Keil.
3. Продемонстрировать работу ЛР на физической плате*.*
4. Оформить отчёт.

**На физической плате присутствует дребезг контактов, от которого необходимо избавиться программным способом.**

Схема в симуляторе Proteus:



### **Задание**

Необходимо разработать программу, которая будет непрерывно генерировать дискретные значения сигнала заданной формы (в зависимости от варианта) и отправлять их на компьютер по интерфейсу USART, модуль USART2. Для отображения сигнала на компьютере можно использовать готовые решения (например, программу [SerialPlot](https://hackaday.io/project/5334-serialplot-realtime-plotting-software), лежит в папке “ПОВС/программы”).

Предусмотреть возможность изменять параметры генерируемого сигнала. Набор параметров определяется вариантом задания (частоту дискретизации изменять не требуется). Текущий изменяемый параметр выбирается с помощью кнопки BTN\_1. Порядковый номер изменяемого параметра отображается с помощью светодиодов (если выбран первый параметр, горит светодиод LED\_1, если выбран второй – LED\_2, и т.д.). Нажатие кнопки BTN\_3 увеличивает значение выбранного параметра на заданный шаг, нажатие на кнопку BTN\_2 уменьшает. Шаг изменения параметра, а также минимальные и максимальные значения выбираются студентом. Отображаемый на компьютере сигнал должен изменять свою форму в реальном времени.

### **Особые требования к реализации**

При работе с модулем USART необходимо использовать механизм прямого доступа к памяти (DMA, Direct Memory Access).

Главный цикл программы не должен содержать постоянного опроса состояния пинов (или постоянной проверки значений переменных) и задержек. Для обработки нажатия кнопок необходимо использовать прерывания. Увеличение номера дискретного значения должно производиться по прерыванию таймера. Обработчики прерываний должны содержать минимум кода (в идеале только помечать, что прерывание произошло). Код, содержащий логику по обработке, должен вызываться из главного цикла. Пока прерываний нет, микропроцессор должен переходить в экономичный режим работы путём выполнения инструкции WFI.

### **Варианты индивидуальных заданий**

**Вариант 1. Синусоидальный сигнал**

|  |  |  |
| --- | --- | --- |

где – амплитуда;

– частота колебаний;

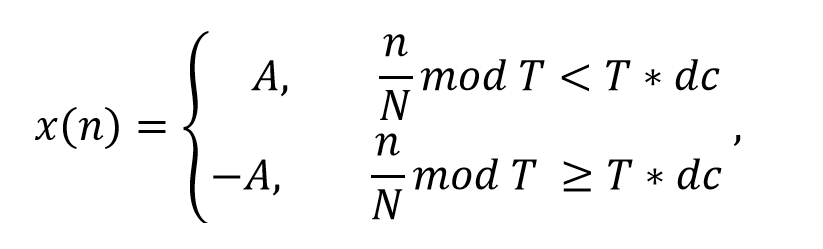
– начальная фаза;

– номер дискретного значения;

– частота дискретизации.

**Вариант 2. Прямоугольный сигнал**

**Вариант 2.1.**



где – амплитуда;

– период колебаний;

– номер дискретного значения;

– частота дискретизации;

– коэффициент заполнения или рабочий цикл.

**Вариант 2.2.**

|  |  |  |
| --- | --- | --- |

где – амплитуда;

– количество гармонических составляющих;

– номер гармонической составляющей;

– частота колебаний;

– номер дискретного значения;

– частота дискретизации.

**Вариант 3. Треугольный сигнал**

**Вариант 3.1.**

|  |  |  |
| --- | --- | --- |

где – амплитуда;

– период колебаний;

– номер дискретного значения;

– частота дискретизации.

**Вариант 3.2.**

|  |  |  |
| --- | --- | --- |

где – амплитуда;

– частота колебаний;

– начальная фаза;

– номер дискретного значения;

– частота дискретизации.

**Вариант 4. Пилообразный сигнал**

**Вариант 4.1.**

|  |  |  |
| --- | --- | --- |

где – амплитуда;

– период колебаний;

– номер дискретного значения;

– частота дискретизации.

**Вариант 4.2.**

|  |  |  |
| --- | --- | --- |

где – амплитуда;

– частота колебаний;

– начальная фаза;

– номер дискретного значения;

– частота дискретизации.