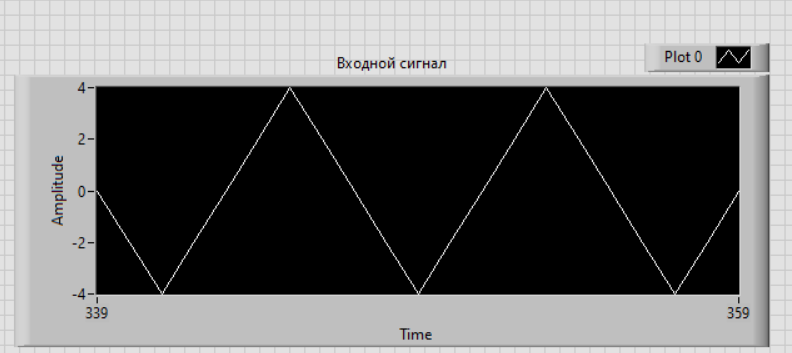
**Задание 3**

**ОБРАБОТКА ДАННЫХ С ТЕКСТОВЫХ ФАЙЛОВ. РАБОТА С ОДНОМЕРНЫМ МАССИВОМ.**

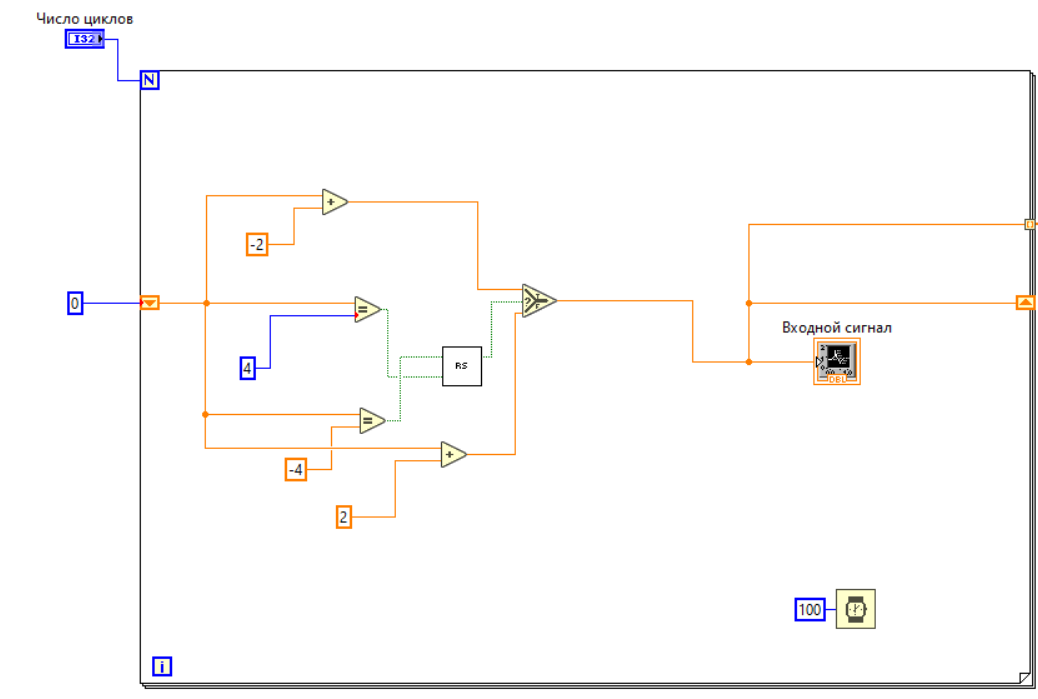
**Задание:** Дан текстовый файл, в который записаны показания датчика, фиксирующего изменения рассматриваемой величины в виде одномерного массива. Необходимо считать данные, вывести на экране график физической величины, определить минимальное и максимальное значение (амплитуду сигнала), а также изменить все отрицательные элементы на положительные и вывести их в другой файл.

**Пример программы:**

В качестве изменяющегося во времени сигнала возьмем пилообразный с амплитудой 4 условные единицы.

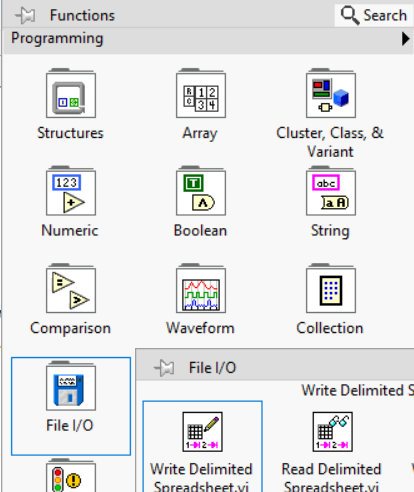
****

Пример программы модулирующей такой сигнал.

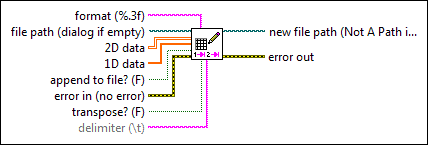
****

Всего будем работать с 20 значениями, поэтому прописываем в числовом контроллере (подключен к N) число 20.

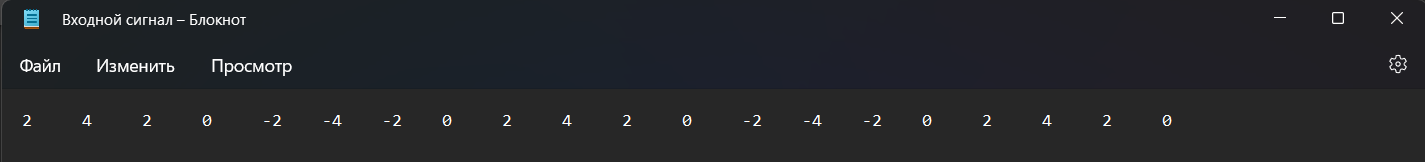
Для записи в файл используем библиотеку File I/O - Write delimited spreadsheet.vi. Рисунок ниже.



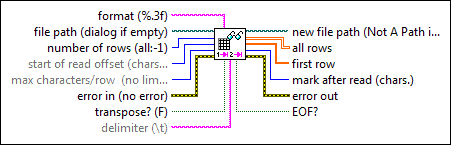
Рассмотрим данный виртуальный прибор. Ниже приведено описание всех входов и выходов, представленное с официального сайта National Instruments.



|  |  |
| --- | --- |
|  | format указывает, как преобразовать числа в символы. Если используется формат %.3f (по умолчанию), ВП создает строку достаточной длины, чтобы содержать число, с тремя цифрами справа от десятичной точки. Если формат равен %d, ВП преобразует данные в целочисленную форму, используя столько символов, сколько необходимо, чтобы вместить все число. Если формат равен %s, ВП копирует входную строку. Используйте синтаксис строки формата. |
|  | путь к файлу — это путь к файлу. Если путь к файлу пуст (по умолчанию) или имеет значение <Not A Path>, ВП отображает диалоговое окно, из которого вы можете выбрать файл. Ошибка 43 возникает, если вы отмените диалоговое окно. |
|  | 2D-данные определяют данные для записи в файл, если 1D-данные не подключены или пусты. |
|  | Данные 1D задают данные для записи в файл, если этот ввод не пуст. Прежде чем продолжить, VI преобразует одномерный массив в двумерный массив. |
|  | добавить в файл? добавляет данные в существующий файл, если TRUE. Если добавить в файл? FALSE (по умолчанию), ВП заменяет данные в существующем файле. Если существующего файла нет, VI создает новый файл. |
|  | error in описывает условия ошибки, возникающие перед запуском этого узла. Этот ввод обеспечивает стандартную ошибку в функциональности. |
|  | транспонировать? указывает, транспонирует ли ВП данные после преобразования их из строки. Значение по умолчанию — ЛОЖЬ. Если транспонировать? FALSE, каждый вызов этого ВП создает новую строку или строку в файле. |
|  | разделитель — это символ или строка символов, которые используются для разделения полей в файле электронной таблицы. Например, значение , (запятая) указывает одну запятую в качестве разделителя. По умолчанию используется \t, который указывает один символ табуляции в качестве разделителя. |
|  | новый путь к файлу возвращает путь к файлу. |
|  | error out содержит информацию об ошибке. Этот вывод обеспечивает стандартную функциональность вывода ошибок.  Подключим данный VI как показано ниже. Ко всем входам подключены стандартные значения (кроме формата, в нем прописано значение %.d, что соответствует целым числа со знаком).    Для выбора пути к файлу нажмите на папку на лицевой панели:    Далее запустим разработанную программу. В результате выполнения в файл запишутся определённые данные, полностью совпадающие с входным сигналом. |

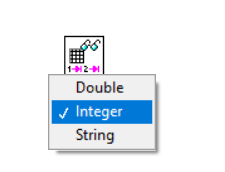


Приступим к чтению значения из файла, для этого будем использовать Read delimited spreadsheet.vi, из библиотеки File I/O. Описание данного виртуального с официального сайта:

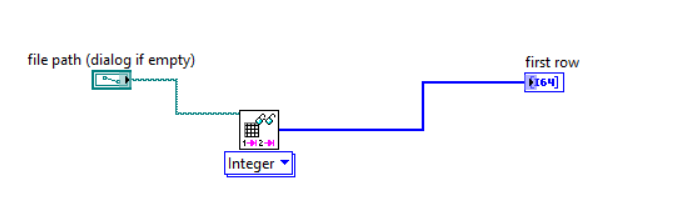


|  |  |
| --- | --- |
|  | format указывает, как преобразовать числа в символы. Если используется формат %.3f (по умолчанию), ВП создает строку достаточной длины, чтобы содержать число, с тремя цифрами справа от десятичной точки. Если формат равен %d, ВП преобразует данные в целочисленную форму, используя столько символов, сколько необходимо, чтобы вместить все число. Если формат равен %s, ВП копирует входную строку. Используйте синтаксис строки формата. |
|  | путь к файлу — это путь к файлу. Если путь к файлу пуст (по умолчанию) или имеет значение <Not A Path>, ВП отображает диалоговое окно, из которого вы можете выбрать файл. Ошибка 43 возникает, если вы отмените диалоговое окно. |
|  | количество строк — это максимальное количество строк или строк, которые считывает ВП. Для этого ВП строка представляет собой строку символов, заканчивающуюся возвратом каретки, переводом строки или возвратом каретки, за которым следует перевод строки; строка, заканчивающаяся в конце файла; или строка с максимальной длиной строки, определяемой максимальным количеством символов на ввод строки. Если количество строк <0, ВП читает весь файл. Значение по умолчанию равно –1. |
|  | Смещение начала чтения задает позицию в файле в символах (байтах), с которой ВП начинает чтение. Единицей смещения являются байты, а не числа, потому что файлы потока байтов могут содержать сегменты данных разных типов. Таким образом, чтобы прочитать массив из 100 чисел, следующий за заголовком из 57 символов, установите смещение начала чтения равным 57. |
|  | max characters/row — это максимальное количество символов, которое ВП считывает перед завершением поиска конца строки или строки. Значение по умолчанию равно 0, что означает, что количество символов, считываемых ВП, не ограничено. |
|  | error in описывает условия ошибки, возникающие перед запуском этого узла. Этот ввод обеспечивает стандартную ошибку в функциональности. |
|  | транспонировать? указывает, транспонирует ли ВП данные после преобразования их из строки. Значение по умолчанию — ЛОЖЬ. |
|  | разделитель — это символ или строка символов, которые используются для разделения полей в файле электронной таблицы. Например, значение , (запятая) указывает одну запятую в качестве разделителя. По умолчанию используется \t, который указывает один символ табуляции в качестве разделителя. |
|  | новый путь к файлу возвращает путь к файлу. |
|  | все строки — это данные, считанные из файла. |
|  | первая строка — это первая строка массива всех строк. Вы можете использовать этот вывод, когда хотите прочитать одну строку в одномерный массив. |
|  | mark after read возвращает символ (байт) в файле, следующий за последним прочитанным символом. |
|  | error out содержит информацию об ошибке. Этот вывод обеспечивает стандартную функциональность вывода ошибок. |
|  | EOF? TRUE, если вы пытаетесь прочитать за конец файла. |

Перед началом подключения установим тип данных, которые мы будем считывать (integer для заданного сигнала).



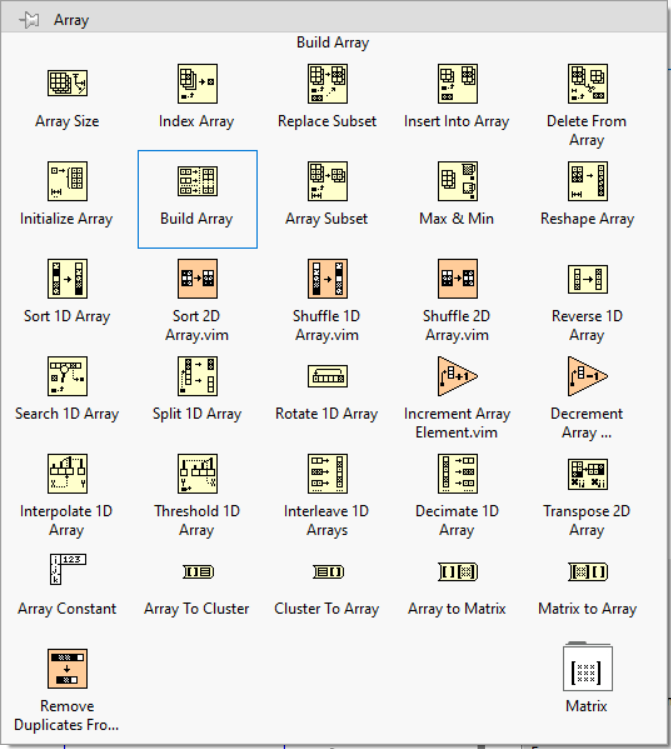
Теперь наконец подключим данный VI. Так как мы использовали стандартные значения на входах, предыдущего VI для записи, необходимость в подключении всех входов отпадает. Поэтому подключение будет таким:



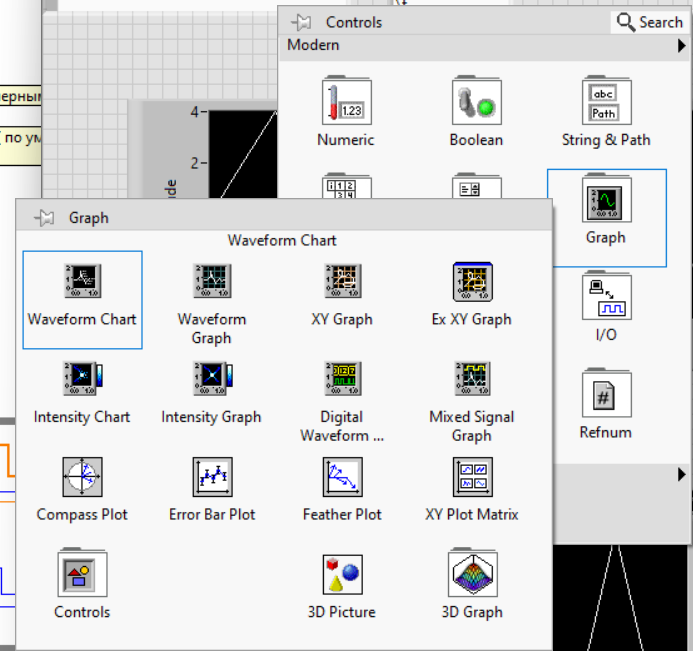
Где в File path укажем местоположение файла входных данных. Мы используем вывод first row (первая строка) только потому, что работает с одномерным массивом, не имеющим большего количества строк. При работе с двумерными массивами необходимо подключиться к all row.

В результате мы получили те же самые данные.

Подсказка: По заданию вам необходимо будет заменить все отрицательные элементы на положительные. Внимательно разберите функции библиотеки Array:



В ней мы будем использовать Insert Into Array и Index Array, а также Max and Min, для обработки полученных данных. Графический вывод осуществляется, используя Waveform Chart, на лицевой панели.



В результате у вас получиться обработанный сигнал. Который нужно будет записать в другой файл. Удачи в выполнении!