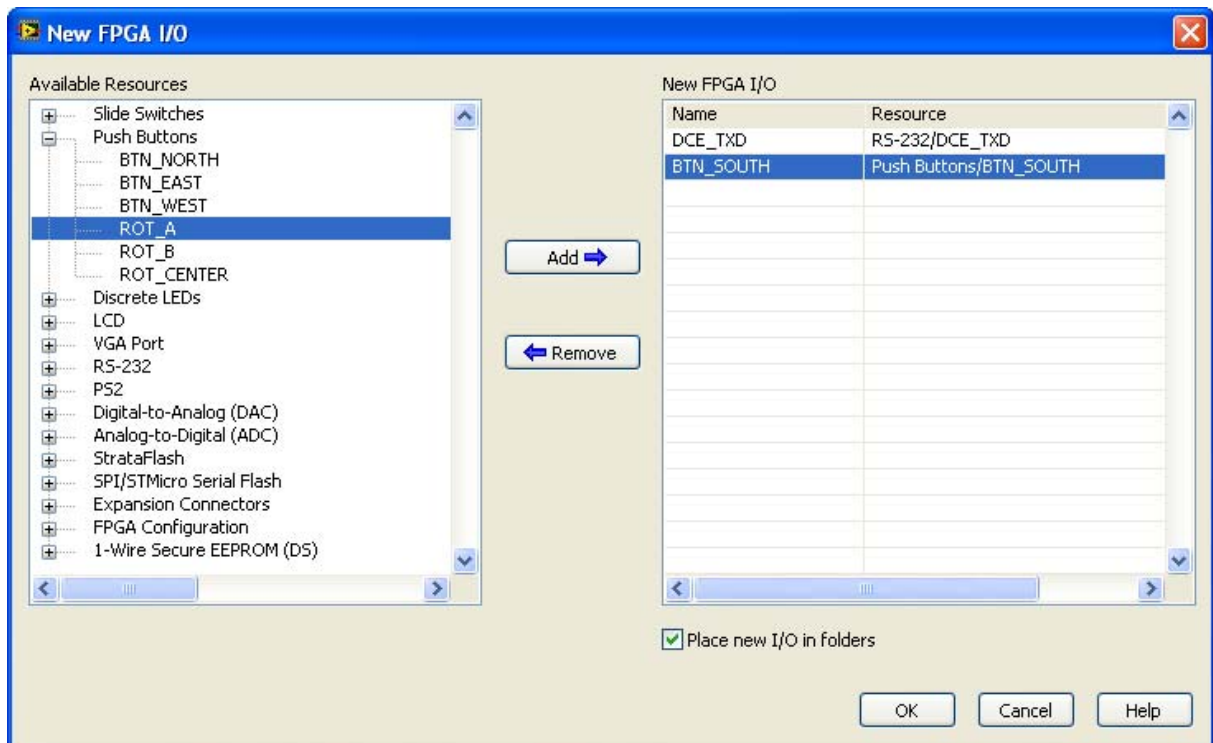


## **Лабораторная работа 2**

**“Реализация последовательной передачи данных в  
LabView FPGA на плате Xilinx SPARTAN-3E”**

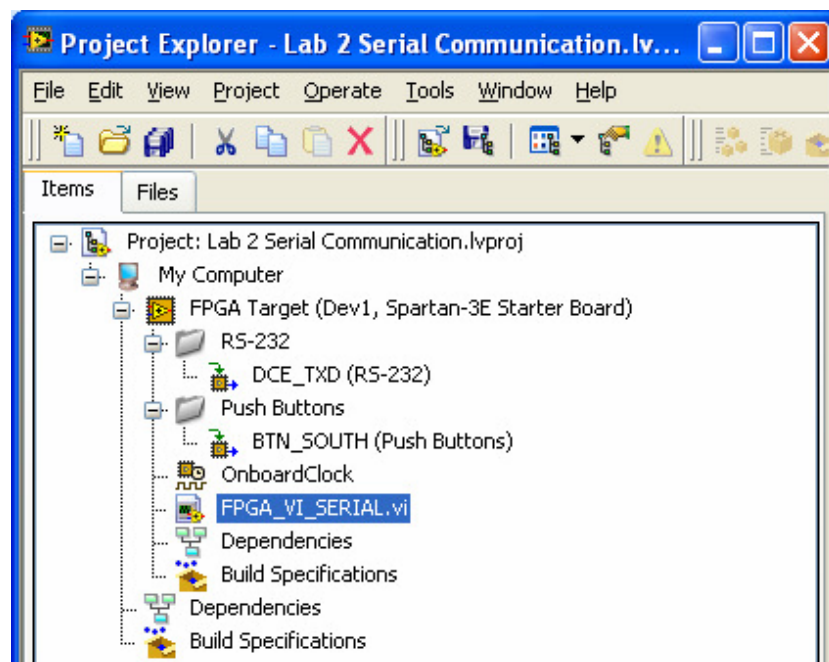
## Шаг 1: Добавление выводов FPGA в проект

Данный шаг был подробно расписан в первой лабораторной работе. В проект необходимо добавить следующие выводы *DCE\_TXD* и *BTN\_SOUTH*.



После выбора выводов, добавьте в проект виртуальный прибор, и присвойте ему имя “FPGA\_VI\_Serial”.

Окно “Project Explorer” должно выглядеть как на рисунке.



## Шаг 2: Краткий обзор протокола RS-232

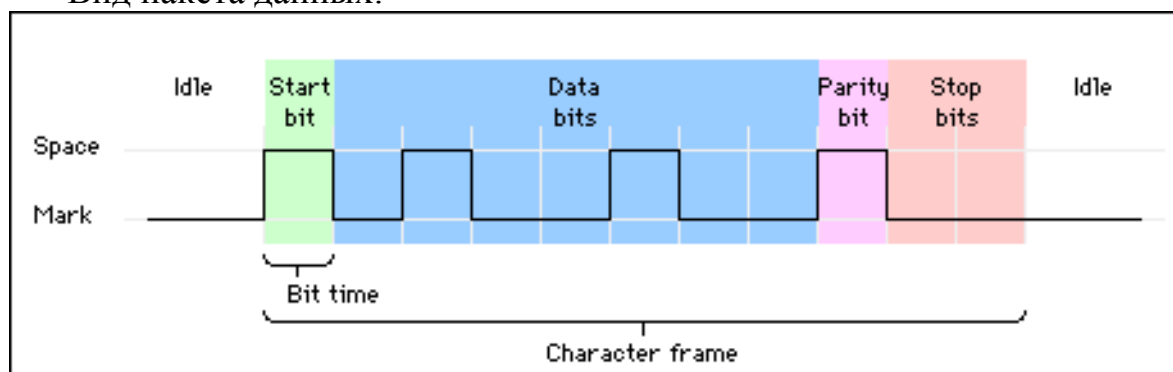
Полезные ссылки:

<http://developer.apple.com/documentation/mac/Devices/Devices-313.html>

<http://digital.ni.com/public.nsf/websearch/2ad81b9060162e708625678c006dfc62>

[http://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver/transmitter](http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter)

Вид пакета данных:



В данной лабораторной работе мы создадим функцию, которая будет посылать набор символов для ПК когда пользователь нажимает на кнопку (*BTN\_SOUTH*). Посылаться будет строка “Hello world!”. Строка храниться в памяти реализованной при помощи таблицы истинности (LUT).

На первом этапе необходимо определить характеристики передачи:

- Скорость – 115200;
- Биты данных – 8;
- Проверка четности – нет;
- Стоббит – 1;
- Управление потоком – аппаратное.

Затем, необходимо перевести строку в набор ASCII символов, для чего необходимо использовать специальные таблицы (или программы), например одна из таких таблиц доступна на сайте: <http://www.asciitable.com/>.

В результате чего, получаем следующий набор

H	e	l	l	o	SPC	w	o	r	l	d	!	CR	LF
72	101	108	108	111	32	87	111	114	108	100	33	13	10

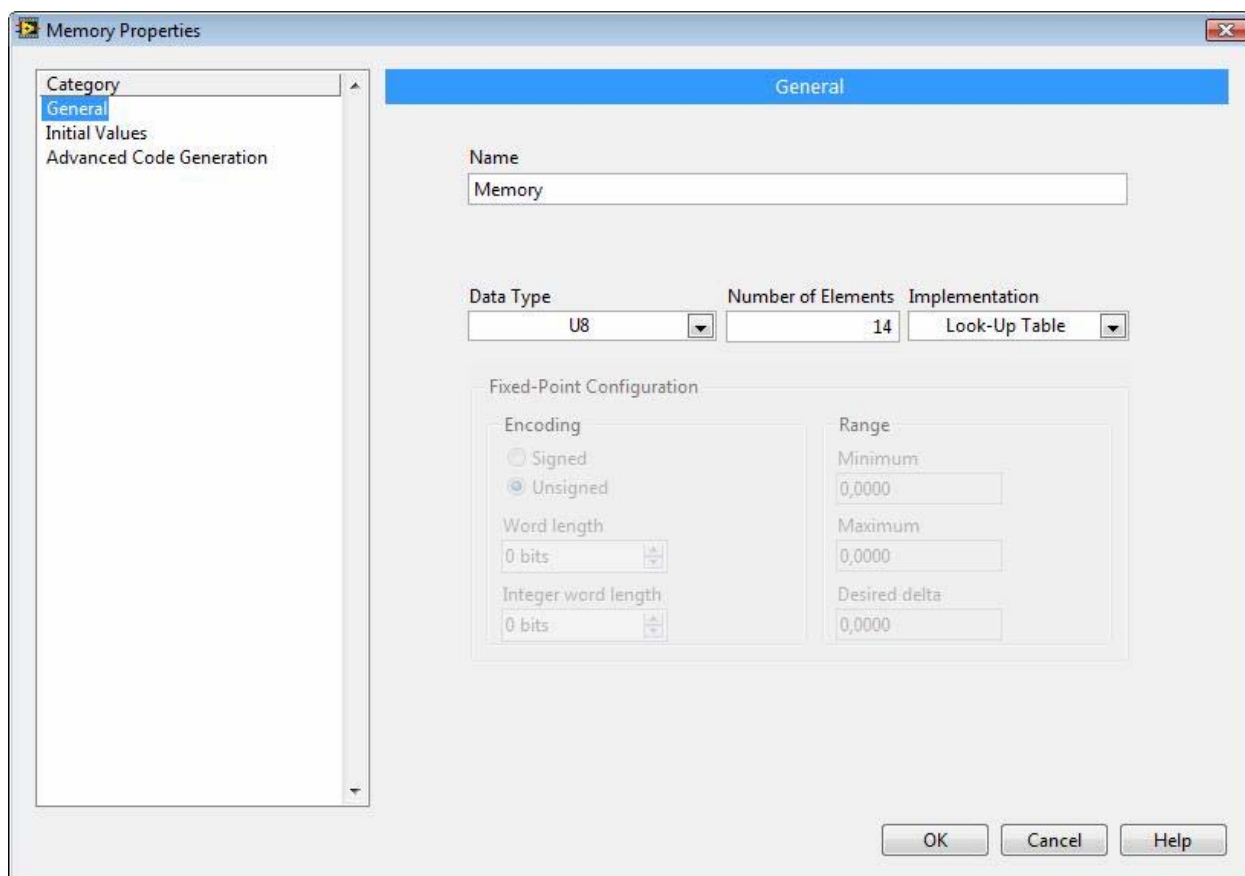
Последние два символа нужны для того, что бы каждое сообщение начиналось в начале новой строки.

### Шаг 3: Реализация

На первом этапе необходимо создать массив данных, который будет передаваться по COM порту. Передаваемая строка будет храниться в памяти. Поместите компонент *Memory* (Из раздела *Memory & Fifo->VI-Defined Memory Configuration*). Вызовите меню настройки двойным щелчком по элементу:

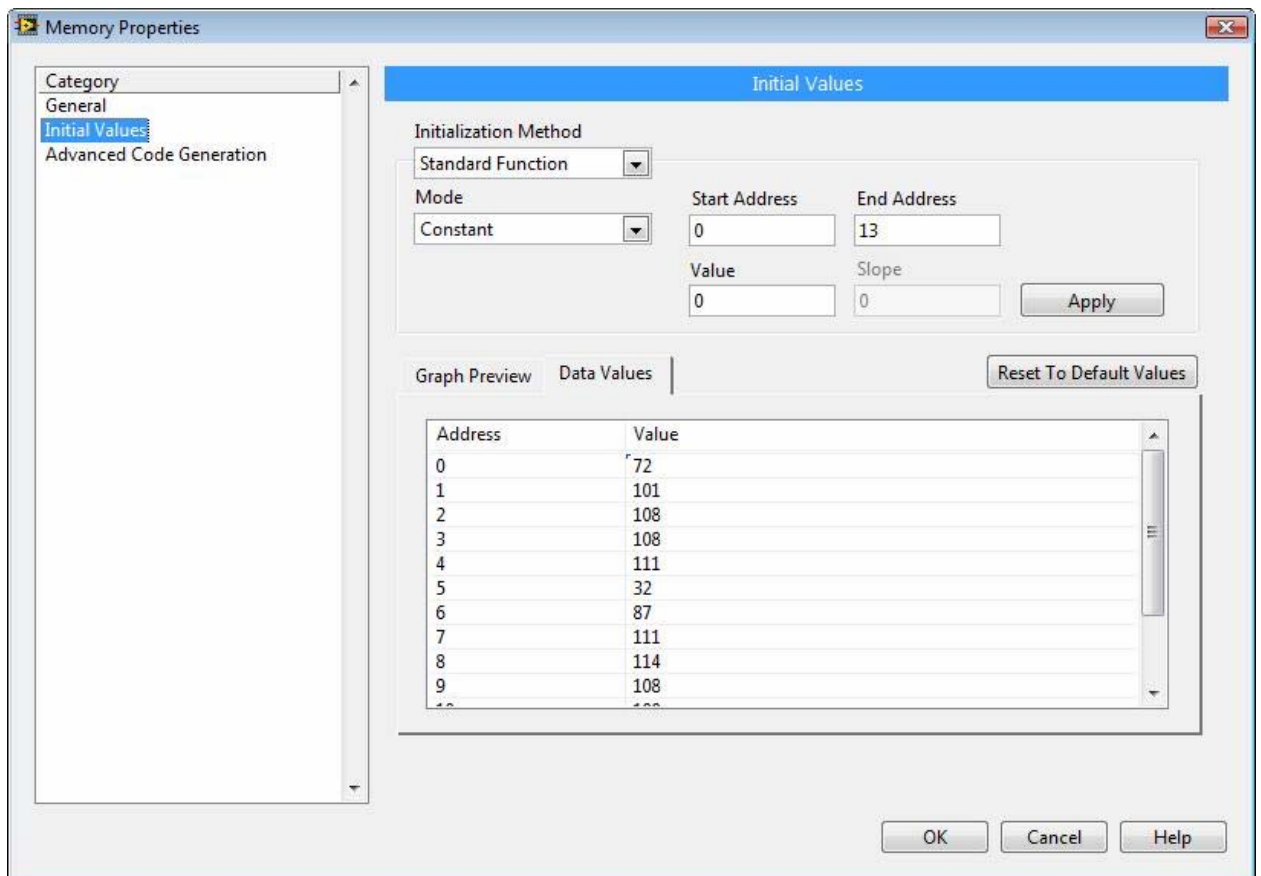


В категории *General* настраивается тип данных (8 бит без знака) хранимых в памяти, количество элементов (14) и способ реализации, а также задается имя компонента. Объем хранимой информации мал, проект не занимает большого объема кристалла и к нему не предъявляются жесткие требования по быстродействию, поэтому выберем реализацию памяти на LUT (Look-Up Table)<sup>1</sup>.



Затем перейдите в категорию инициализации. Перед тем, как приступить к заполнению памяти, нажмите кнопку *Apply* и выберите закладку *Data Values*. Заполните столбец *Value*, и нажмите *Ok*.

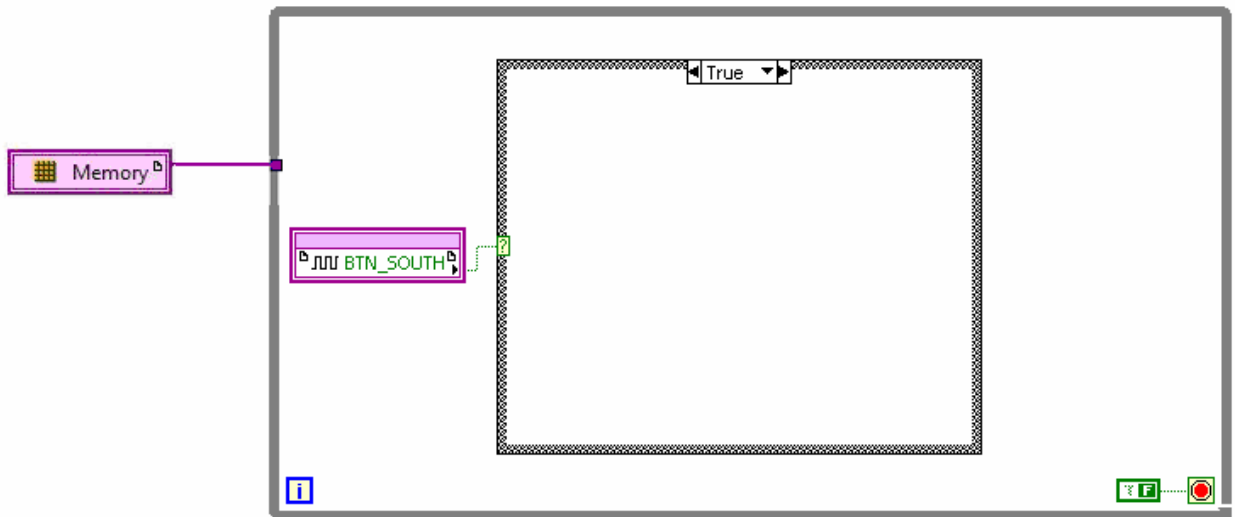
<sup>1</sup> Кроме того, компиляция при использовании Блочной памяти невозможно, в связи с ошибками в драйвере для платы SPARTAN-3E модуля LabView FPGA. По возможности, всегда следует использовать блочную память для хранения информации в ПЛИС.



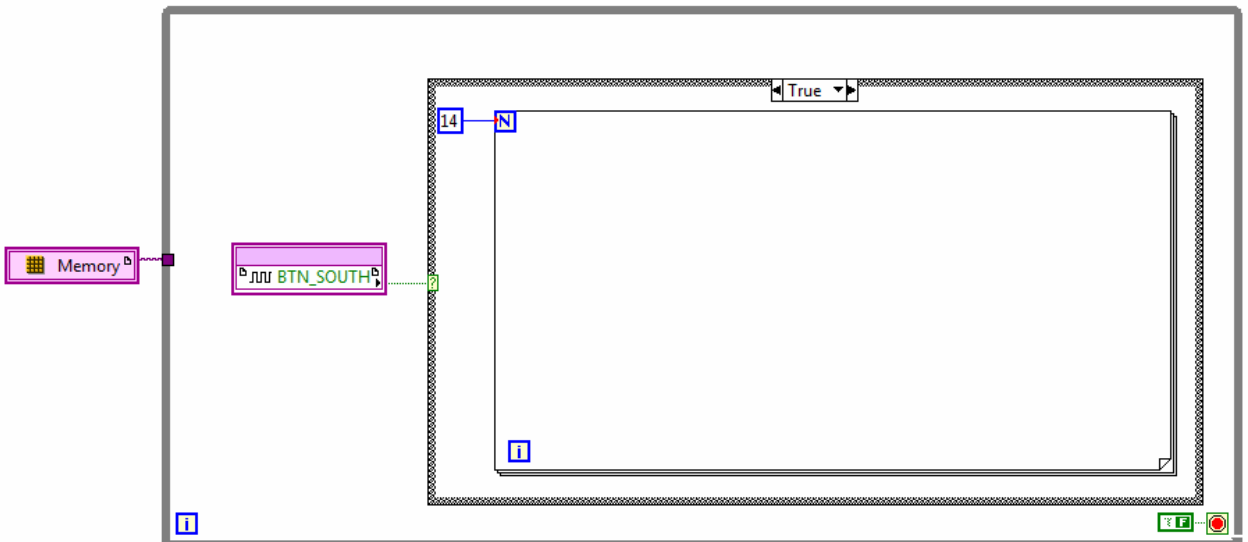
Создайте главный цикл программы, при этом элемент *Memory* должен находиться вне этого цикла, так как дальнейшая работа с ним ведется при помощи компонента *Memory Method Node*.



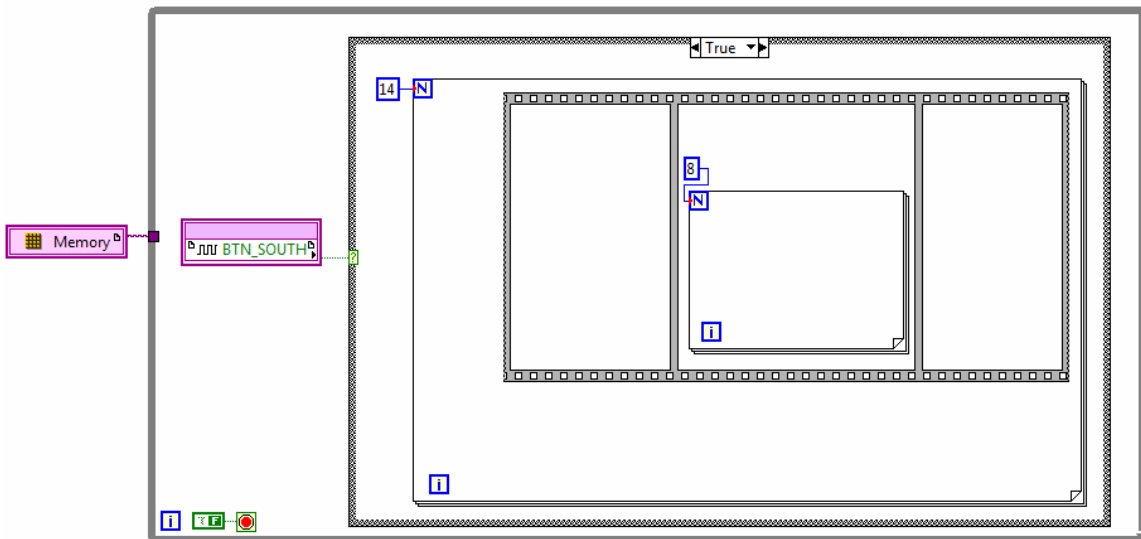
Как уже отмечалось выше, передача данных должна осуществляться, только при нажатии кнопки *BTN\_SOUTH*, поэтому внутри цикла необходимо разместить структуру *Case*, управление которой будет осуществляться при помощи кнопки.



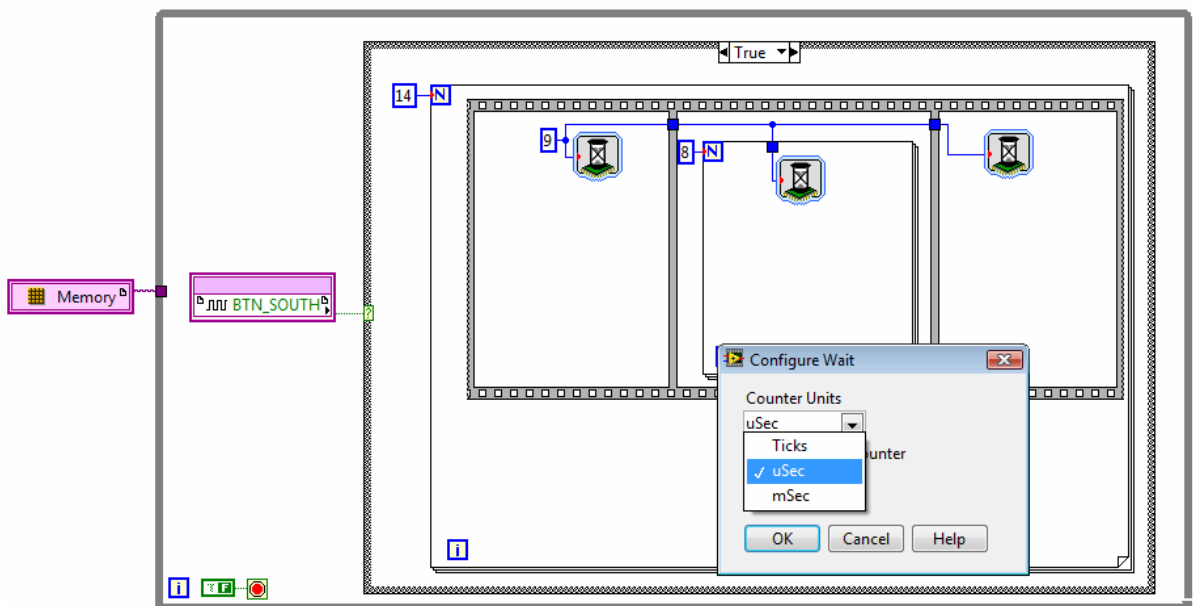
Поместите структуру *For Loop* в “*True case*”, параметру N подключите константу 14.



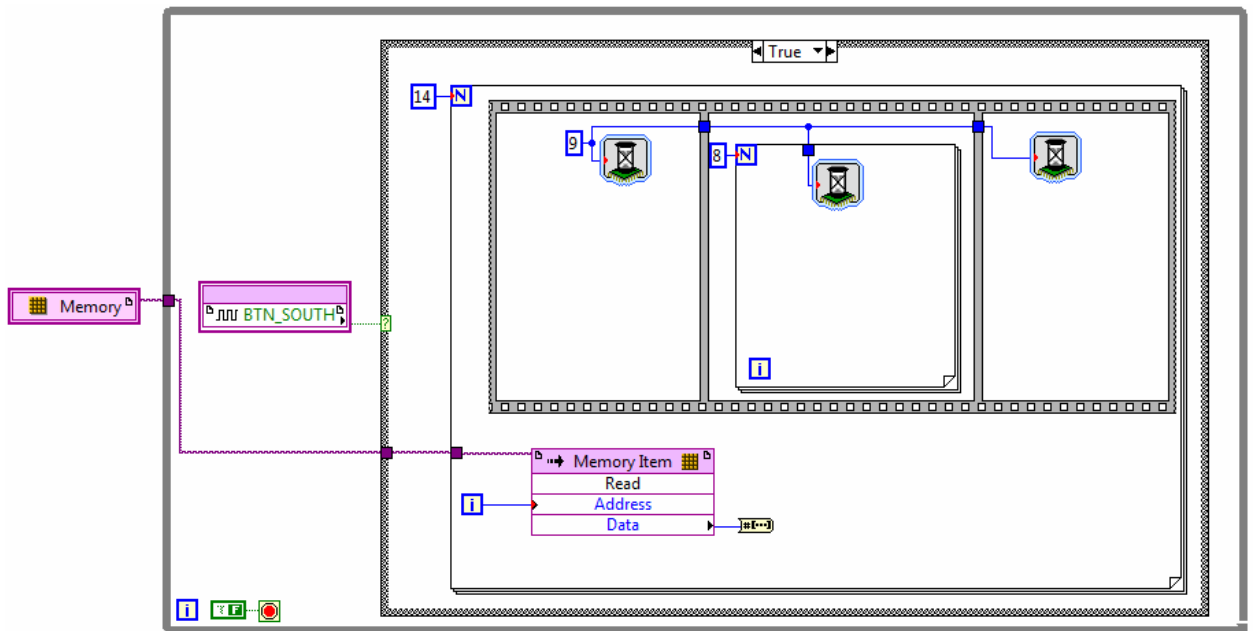
Теперь перейдем непосредственно к реализации протокола RS-232. Для этого поместите *Sequence Structure*, состоящий из трех кадров, в созданном цикле. Во втором кадре разместите еще одну структуру *For Loop*, и подключите к параметру N константу 8.



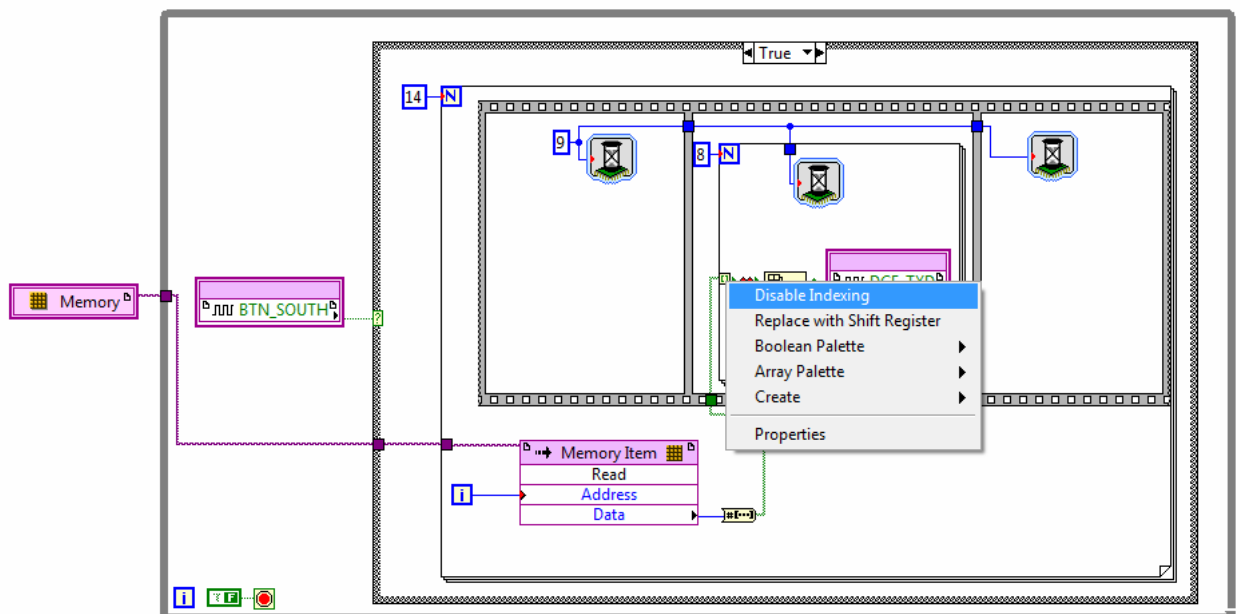
Для передачи на скорости 115200 бит в секунду, каждый бит должен передаваться в течение  $1/115200$  секунды, то есть примерно 9 мс. Поместите в каждый кадр задержку (во втором кадре задержка должна быть внутри цикла *For*), задав единицу счета **uSec**. И подключите к каждому элементу константу 9.



Теперь необходимо реализовать чтение ASCII кода из памяти и перевода его в двоичную последовательность. Для этого поместим функцию *Memory Method Node* в первый цикл *For loop*, и установим параметр **Read**. К входу **Address** необходимо подключить счетчик итераций **i** структуры *For loop*. Поле **Data** соединим с входом функции *Number to Binary Array*.



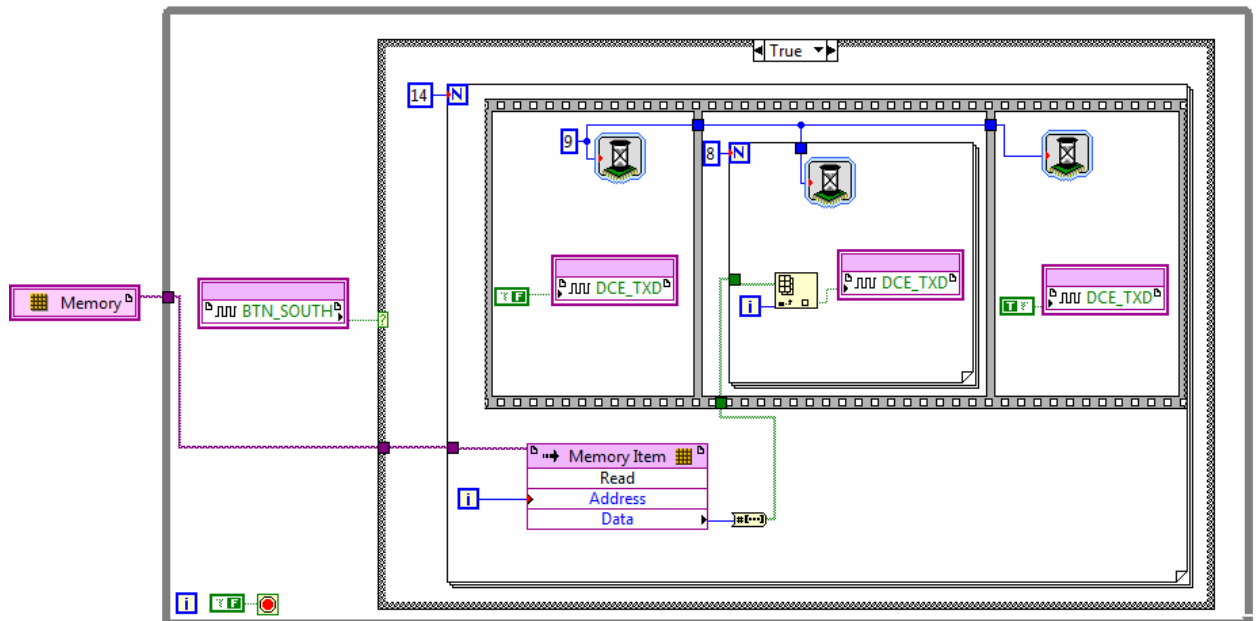
На выходе получим массив двоичных чисел. Для того что бы ее передать, необходимо воспользоваться функцией *Index Array*, поместив ее во второй цикл *For Loop*. На вход **index** необходимо подавать значения со счетчика итераций **i**. Выход функции подключается к *I/O Node*, в качестве объекта которой установлен *DCE\_TXD*. Для устранения возникшей ошибки, вызовете выпадающее меню на точке входа в цикл *For* и выберите *Disable Indexing*.



Обратимся вновь к диаграмме передачи данных. Мы уже реализовали часть пакета, которая передает данные. Теперь необходимо добавить формирование битов начала и конца пакета данных. Start bit характеризуется низким уровнем на линии передачи. Для его формирования необходимо в первом кадре разместить *I/O Node DCE\_TXD*, и на вход подключить константу



False. Stop bit характеризуется высоким уровнем на линии передачи. Parity bit в реализуемом нами протоколе отсутствует.



На этом разработка устройства окончена. Запустить проект на компиляцию и прошивку, после чего откройте программу Terminal. Выберите номер порта, к которому подключена Ваша плата, введите верные настройки передачи данных. Если все было сделано правильно, то при нажатии на кнопку SOUTH на плате, в окне терминала будет выводиться строка “Hello world!”.