

# Дополнение к работе 2. Модуль ввода

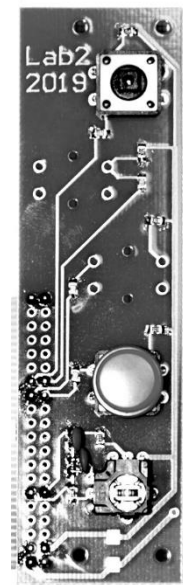
Прочитайте основное описание работы 2 до практического задания 1 (стр. 18). Само задание не выполняйте.

Модуль ввода содержит один двухкоординатный джойстик и 4 кнопки.

Кнопка	Порт	Линия магистрали Avalon
S1 (верхняя)	Port 1, bit 4	AD12
S2	Port 1, bit 5	AD13
S3	Port 1, bit 6	AD14
S4 (нижняя)	Port 1, bit 7	AD15

Когда кнопка отпущена, на соответствующую линию подаётся +5 В (логическая 1) через притягивающий резистор номиналом 10 кОм; нажатая кнопка притягивает линию к земле (логический 0) через токоограничительный резистор 1 кОм. Благодаря этим резисторам модуль ввода не должен мешать работе других устройств магистрали Avalon.

Имеется тестовая программа TST-lab02-2019, позволяющая проверить работу оборудования. В результате выполнения данной работы вы должны создать нечто подобное.



## Задание 1-1. Отображения состояния кнопок на индикаторе магистрали

1. Создайте новый проект из шаблона User Interface Application(TSANI).
2. Переведите модуль индикатора магистрали в режим отображения состояния линий магистрали: `portMask(2, 0x07); portOut(2, 0x01);`
3. Переведите порты 0 и 1 в режим чтения: `portMask(0, 0x00); portMask(1, 0x00);`

После выполнения этих действий индикаторы AD0...11 должны оказаться погашенными, а AD12...15 горящими; при нажатии на кнопку модуля ввода соответствующий индикатор должен погаснуть.

## Задание 1-2. Чтение состояния кнопок через порт

Добавьте в программу таймер. В его функции обратного вызова читайте порт 1. Если прочитанное значение отличается от предыдущего (заведите глобальную переменную, чтобы запоминать предыдущее состояние), выводите его на экран.

## Задание 1-3. Наблюдение дребезга

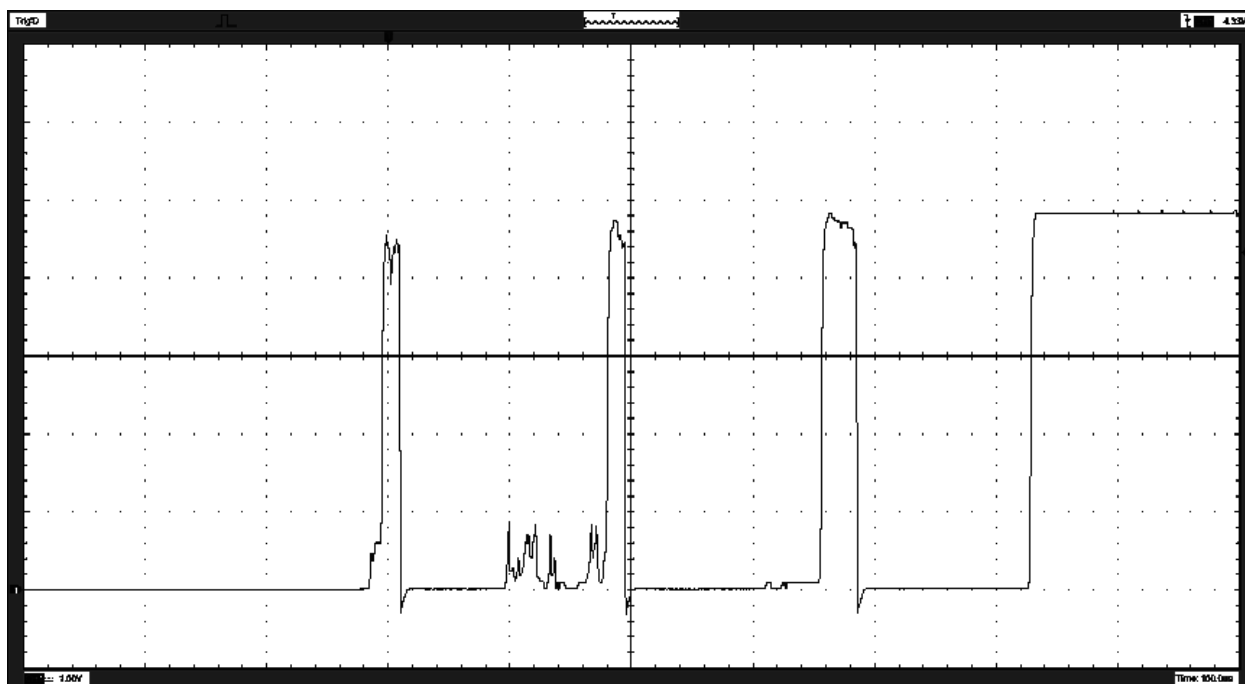
Добавьте к выводу на экран номер переключения (от начала работы программы) и время, прошедшее с предыдущего изменения состояния порта. Воспользуйтесь функцией `double Timer (void)`, она возвращает время в секундах.

Сделайте свою программу максимально быстрой. Для этого:

- Не используйте графический интерфейс (вывод на экран осуществляйте функцией `printf()`, вместо таймера используйте бесконечный цикл, весь код должен располагаться в функции `main()` до вызова функции `RunUserInterface()`)
- Переведите проект в режим Release (Build → Configuration)
- Переведите проект в режим консольного приложения (Build → Target Settings... → +Create console application)

Несколько (десятков) раз нажмите и отпустите кнопку, стараясь делать это через одинаковые интервалы времени и считая количество переключений. Какое количество переключений насчитала программа? Если такое же — продолжите эксперимент. Если больше — изучите вывод программы. Обратите внимание на моменты, когда время переключения неправдоподобно малое (порядка миллисекунды и меньше).

Имеющееся оборудование ввиду относительно невысокого быстродействия не позволяет в полной мере изучить явление дребезга контактов. Рассмотрите осциллограмму длительностью 1 мс. Если бы ваша программа была столь же быстрой, сколько переключений она бы насчитала? Придумайте алгоритм борьбы с дребезгом.



## Задание 1-4. Работа с портом вывода

Закомментируйте код от заданий 1-2 и 1-3 (или создайте новый проект и повторите задание 1-1).

Оставив старшие 4 линии порта 1 в режиме ввода, переведите младшие 4 линии в режим вывода.

Добавьте в программу таймер. В его функции обратного вызова:

1. Прочитайте регистр чтения порта 1
2. В полученном 8-битном числе сделайте бит 0 равным биту 7; бит 1 — инвертированному биту 7; остальные биты оставьте неизменными
3. Запишите результат в регистр записи порта 1

**Выполните задание 1 из основного описания работы 2.**

**Прочитайте раздел «ЦАП и АЦП NI PXI-6251» из основного описания работы 2 (стр. 19).**

**Практическое задание 2 не выполняйте.**

В верхней части терминального блока имеется аналоговый вольтметр постоянного тока с пределами измерения от 0 до +10 В. Его можно подключить к выходам А00 или А01 ЦАП модуля NI PXI-6251. Будьте осторожны: не подавайте на вольтметр отрицательное напряжение.

Двухосевой джойстик, расположенный в нижней части модуля ввода, по сути является парой переменных резисторов. На каждый резистор подаётся постоянное напряжение, а средние точки подключены к аналоговым входам 2 и 3 АЦП модуля NI PXI-6251.

## Задание 2-1. Работа с АЦП модуля NI PXI-6251

Создайте новый проект. Добавьте в графический интерфейс два поля вывода чисел (Numeric → Numeric). Добавьте таймер. В функции обратного вызова таймера читайте входы 2 и 3 АЦП и выводите полученные значения в поля.

С помощью полученной программы исследуйте джойстик:

- Какая ось (X/Y) соответствует какому входу АЦП?
- Какое направление (вверх/вниз, влево/вправо) каждой оси соответствует увеличению/уменьшению напряжения?
- В каких пределах меняется напряжение?
- Какое напряжение выдаёт джойстик без внешнего воздействия?

## Задание 2-2. Работа с ЦАП модуля NI PXI-6251

Продублируйте напряжения, выдаваемые джойстиком, на выходах 0 и 1 ЦАП. Проверьте работу программы с помощью аналогового вольтметра (достаточно проверить лишь один выход ЦАП).

## Задание 2-3. Калибровка джойстика

Сделайте такое линейное преобразование над каждой осью джойстика, чтобы в крайнем левом/нижнем положении получалось число -2,5, а в крайнем правом/верхнем — +2,5.

Добавьте в графический интерфейс элемент Graph → Graph, установите пределы по каждой оси от -3 до +3.

Выводите на график две точки (разными цветами): истинное (после линейного преобразования) положение джойстика и его же, но округлённое до целых.

**Практическое задание 2 основного описания работы 2 можно не выполнять. Продолжите работу с раздела «Специализированная магистраль Avalon» (стр. 20).**

Зорин Артем Викторович

+7 952 949-41-15

[a.v.zorin@inp.nsk.su](mailto:a.v.zorin@inp.nsk.su)

<http://www.caesarion.ru/tsani>