

NXT Camera v3.0

Введение

На этом занятии:

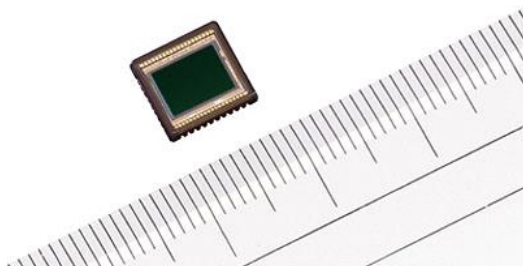
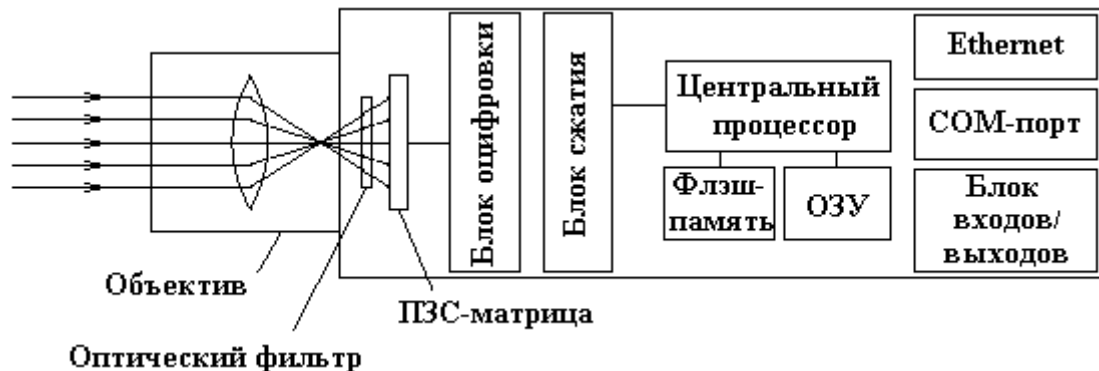
1. Познакомимся с видео датчиком NXT Camera v3.0
2. Научимся конфигурировать видеодатчик при помощи программы NXTCamView
3. Напишем программу подсчета количества объектов определенного цвета, в области зрения видеодатчика

Принцип работы видео датчиков

Устройство и принцип работы камеры

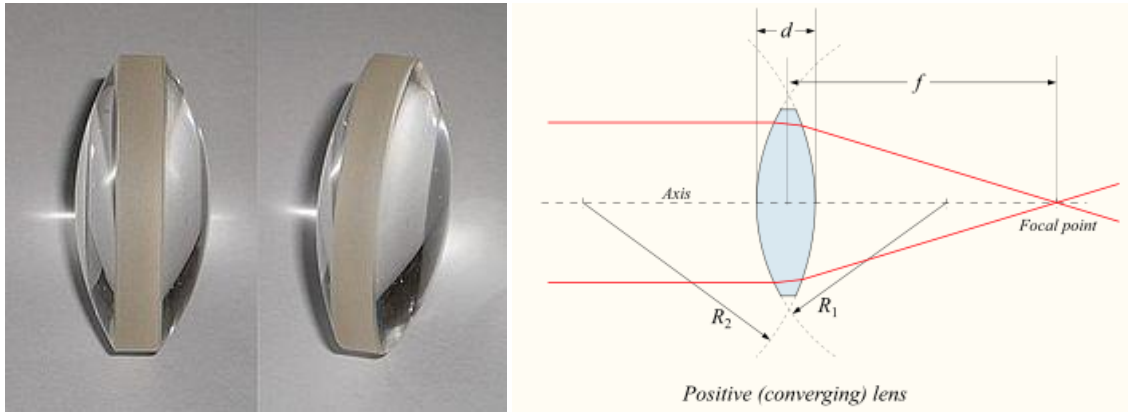
Современная камера представляет собой цифровое устройство, производящее видеосъемку, перобразование аналогового видеосигнала в цифровой, сжатие цифрового видеосигнала и передачу видеоизображения. Поэтому в состав камеры входят следующие компоненты:

- ПЗС-матрица,
- объектив,
- оптический фильтр,
- плата видеозахвата,
- *блок компрессии (сжатия) видеоизображения (не обязательны компонент),*
- центральный процессор,
- ОЗУ,
- *флэш-память (не обязательны компонент),,*
- сетевой интерфейс



В качестве фотоприемника в большинстве камер применяется **ПЗС-матрица** (ПЗС, CCD – прибор с зарядовой связью) – прямоугольная светочувствительная полупроводниковая пластинка с отношением

сторон 3 : 4, которая преобразует падающий на нее свет в электрический сигнал. ПЗС-матрица состоит из большого числа светочувствительных ячеек. Для того чтобы повысить световую чувствительность ПЗС-матрицы, нередко формируют структуру, которая создает микролинзу перед каждой из ячеек. В технических параметрах камеры обычно указывают формат ПЗС-матрицы (длина диагонали матрицы в дюймах), число эффективных пикселей, тип развертки (построчная или чересстрочная) и чувствительность.



Объектив – это линзовая система, предназначенная для проецирования изображения объекта наблюдения на светочувствительный элемент камеры. Объектив является неотъемлемой частью камеры, поэтому от правильности его выбора и установки зависит качество видеоизображения, получаемого камерой. Достаточно часто камера комплектуется объективом. Объективы характеризуются рядом важнейших параметров, таких как фокусное расстояние, относительное отверстие (F), глубина резкости, тип крепления (C , CS), формат.



Оптические инфракрасные отсекающие фильтры, которые устанавливают в камеры, представляют собой оптически точные плоскопараллельные пластинки, монтируемые сверху ПЗС-матрицы. Они работают как оптические низкочастотные фильтры с частотой среза около 700 нм, вблизи красного цвета. Они отсекают инфракрасную составляющую световых волн, обеспечивая камере правильную цветопередачу. Однако, на многие черно-белые камеры такие фильтры не устанавливают, благодаря чему монохромные камеры имеют более высокую чувствительность.

Плата видеозахвата камеры (блок оцифровки) осуществляет преобразование аналогового электрического сигнала, сформированного ПЗС-матрицей, в цифровой формат. Процесс преобразования сигнала состоит из трех этапов:

- Дискретизация,
- Квантование,
- Кодирование.

Дискретизация – считывание амплитуды электрического сигнала через равные промежутки времени (период). Этот этап преобразования сигнала характеризуется частотой дискретизации.

Квантование – это процесс представления результатов дискретизации в цифровой форме. Изменение уровня электрического сигнала за период дискретизации представляется в виде кодового слова из 8, 10 или 12 бит, которые дают соответственно 256, 1024 и 4096 уровней квантования. От числа уровней квантования зависит точность представления сигнала в цифровой форме.

Кодирование. Помимо информации об изменении уровня сигнала, полученной на предыдущем этапе, в процессе кодирования формируются биты, сообщающие о конце синхроимпульса и начале нового кадра, а также дополнительные биты защиты от ошибок.

Блок компрессии камеры выполняет сжатие оцифрованного видеосигнала в один из форматов сжатия (JPEG, MJPEG, MPEG-1/2/4, Wavelet). Благодаря сжатию, сокращается размер видеокadra. Это необходимо для хранения и передачи видеоизображения по сети. Если локальная сеть, к которой подсоединена камера, имеет ограниченную полосу пропускания, то во избежание переполнения сетевого трафика целесообразно сокращать объем передаваемой информации, снизив либо частоту передачи кадров по сети, либо разрешение кадров. Большинство форматов сжатия, которые используют веб камеры, обеспечивает разумный компромисс между этими двумя способами решения проблемы передачи видео по сети. Известные на сегодняшний день форматы сжатия позволяют получить оцифрованный поток с полосой пропускания 64 Кб – 2 Мб (при такой полосе пропускания потоки видеоданных могут работать параллельно с другими потоками данных в сетях).

Сжатие видеоизображения в камере может быть представлено как аппаратно, так и программно. Программная реализация компрессии дешевле, однако из-за высокой вычислительной емкости алгоритмов сжатия она малоэффективна, особенно когда требуется просматривать видеоизображение с камеры в online режиме. Поэтому большинство ведущих производителей выпускают камеры с аппаратной реализацией сжатия. Например, каждая сетевая камера компании AXIS Communications оснащена процессором компрессии ARTPEC, осуществляющим высокоскоростное сжатие видеоизображения в формат JPEG/MJPEG.

Центральный процессор является вычислительным ядром камеры. Он осуществляет операции по выводу оцифрованного и сжатого видеоизображения, а также отвечает за выполнение функций встроенного сервера и управляющей программы для камер.

Интерфейс служит для подключения камеры к компьютеру или к другим устройствам способным работать с камерой.

Карта флэш-памяти позволяет обновлять управляющие программы и конфигурацию камеры.

ОЗУ служит для хранения временных данных, которые генерируются при выполнении управляющих программ и пользовательских скриптов. Многие интернет-камеры имеют так называемый видеобуфер. Это часть ОЗУ, зарезервированная для записи и временного хранения снятых камерой видеокadров. Информация в видеобуфере обновляется циклически, т.е. новый кадр записывается вместо самого старого. Эта функция необходима, если камера выполняет охранное видеонаблюдение, поскольку позволяет восстанавливать события, предшествующие и следующие за сигналом тревоги с подключенных к камере охранных датчиков.

Программа конфигурации NXTCamView

Видео датчик NXT Camera v3.0, далее будем называть его камерой, в своем составе имеет процессор, занимающийся обработкой изображения с матриц датчика, такое устройство называется видео граббером, и в более мощных системах машинного зрения, обычно выполнено в виде отдельной платы. При подключении к NXT, от камеры мы получаем не саму картинку, а результат обработки, это позволяет повысить скорость обработки видео до 30 кадров в секунду.

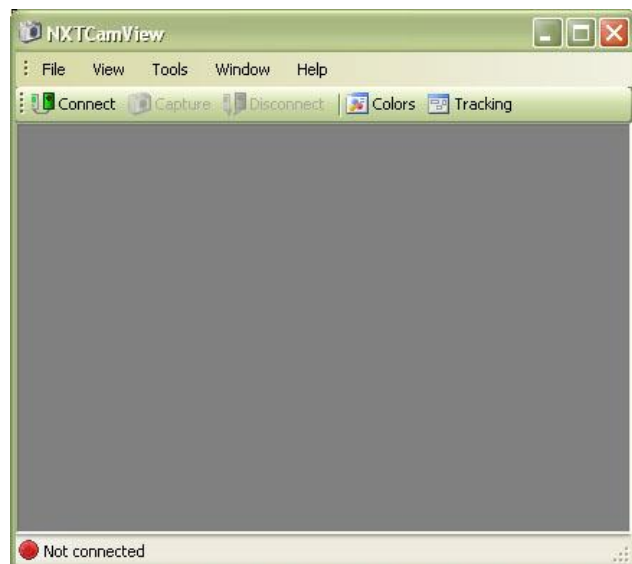
Пакет данных который передает контроллер камеры в NXT состоит из:

- Количества найденных объектов, максимум 8.
Каждый объект представляется в контроллере как прямоугольник определенного размера.
- Данные по каждому объекту
 - Координаты (x, y) для верхнего левого угла прямоугольника;
 - Координаты (x, y) для нижнего правого угла прямоугольника;
 - Цвет объекта.

Для того чтобы контроллер камеры передавал информацию только о нужных нам объектах в кадре, его необходимо сконфигурировать, то есть определить параметры по которым он определяет объекты.

Основной параметр в нашем случае, это цвет объекта.

Открываем программу NXTCamView, меню ПУСК -> **NXTCamView** -> **NXTCamView**.



На панели под меню нам доступны три кнопки, первая - **Connect** для соединения с камерой, вторая - **Colors** настройка палитры цветов для определения объектов, и третья – **Tracking** – запуск работы камеры в режиме обработки изображения, с отображением результатов в окне программы **NXTCamView** на компьютере.

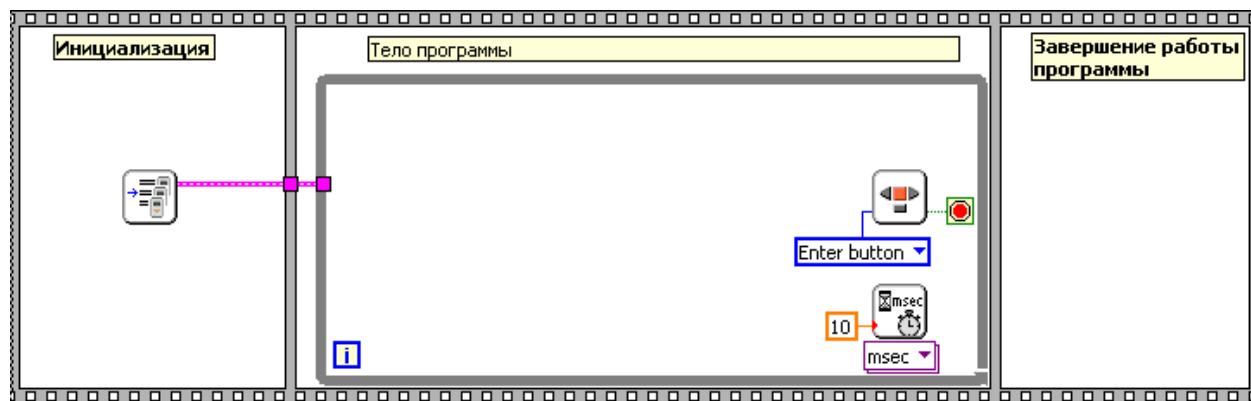
Первая программа с видео датчиком NXT Camera v3.0

Задача:

Подсчет количества квадратных конфет двух цветов на листе бумаги, и отображение их количества на экране монитора.

Программу будем писать для выполнения на компьютере.

Создаем шаблон программы:



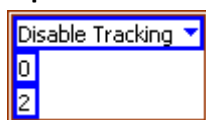
Для работы с видео датчиком **NXT Camera v 3.0** должна быть установлена библиотека **Mindsensors**. Если ее нет, то последующие действия сделать не удастся. Скачать саму библиотеку и инструкцию по ее установке можно с сайта <http://www.mindsensors.com>.

Добавляем функцию для работы с датчиками mindsensors, **Functions -> mindsensors device ->**

mindsensors device  **Acceleration Sensor**  **NXTCam**. В меню функции выбираем **Sensor -> NXTCam**.

Данная функция позволяет управлять работой камеры и имеет пять терминалов для подключения, три входных и два выходных.

- Входные терминалы
 - **Input Port** сверху функции – на данный вход подключается константа с портом, к которому подключена камера
 - **NXT** – терминал подключения кластера с параметрами NXT
 - **Input of NXTCam** – входной кластер с командой и набором параметров для работы камеры



- Первый элемент кластера – команда камере о выполняемой операции

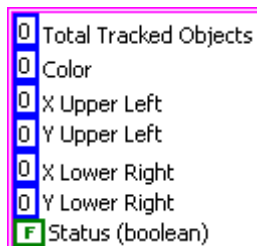
Список интересующих нас команд:

- **Disable Tracking** - Отключить работу камеры
- **Get First Object** – получить информацию об первом объекте
- **Enable Tracking** – Запустить работу камеры на отслеживание объектов
- **Sort Objects by size** – Отсортировать объекты текущего кадра по размеру
- **Sort Objects by Colors** – Отсортировать объекты текущего кадра по цвету
- Второй элемент кластера – номер объекта о котором хотим получить информацию
- Третий элемент кластера – внутренний адрес для команды

- Выходные терминалы
 - **NXT** – терминал для подключения кластера с параметрами NXT
 - **Output of NXTCam** – кластер с данными об объектах обработанных камерой

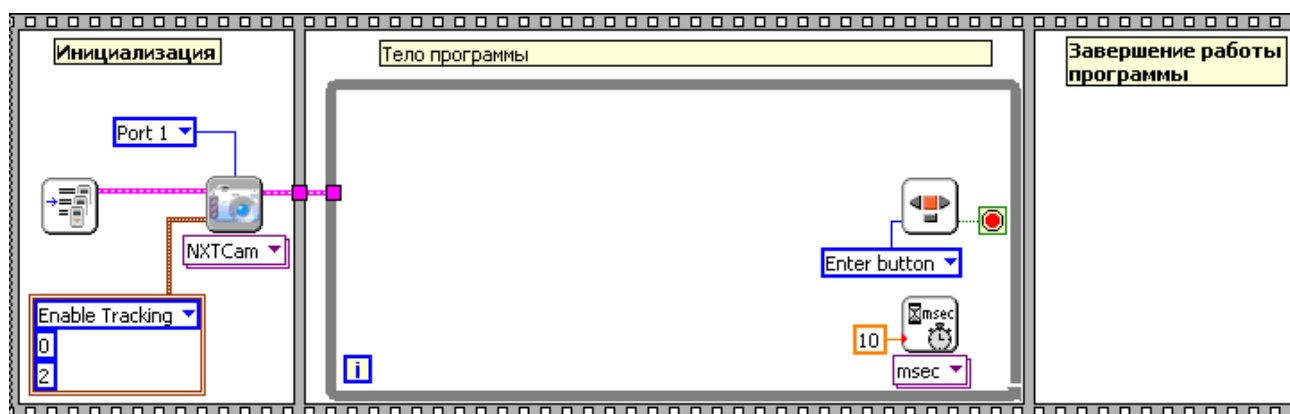


для удобства включим отображение названий у каждого элемента в кластере, клик правой кнопкой мышки на элементе в меню **Visible Items->Label**



- **Total Tracked Objects** – количество найденных объектов при обработке текущего кадра, меняется от 0 до 8.
- **Color** –цвет объекта, номер которого мы указали во входном кластере. При конфигурации камеры в программе CAMView мы указывали палитру из 8 цветовых окон для распознавания объектов, так значение параметра Color соответствует номеру цветового окна в которое попал данный объект и лежит в пределах от 1 до 8.
- **X Upper Left** – координата X левого верхнего угла объекта, номер которого мы указали во входном кластере.
- **Y Upper Left** – координата Y левого верхнего угла объекта, номер которого мы указали во входном кластере.
- **X Lower Left** – координата X правого нижнего угла объекта, номер которого мы указали во входном кластере.
- **Y Lower Left** – координата Y правого нижнего угла объекта, номер которого мы указали во входном кластере.

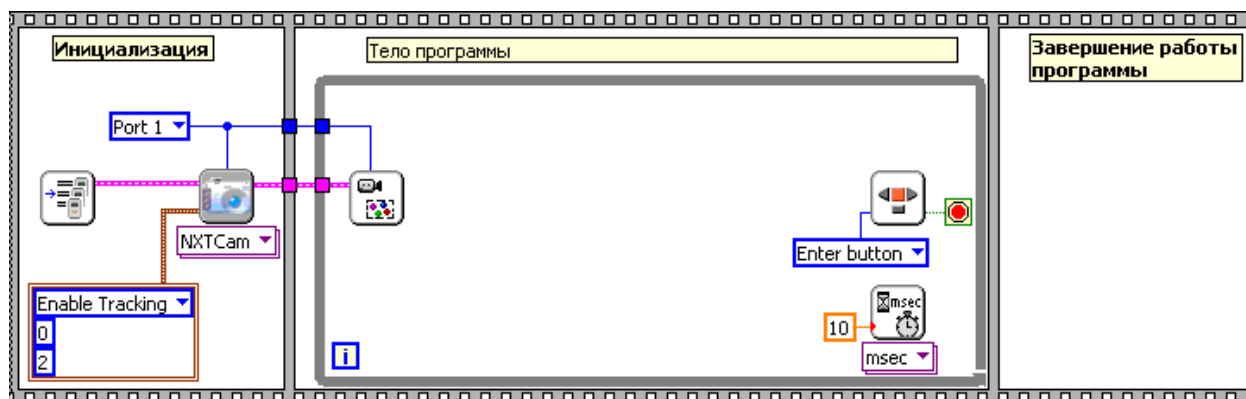
Вернемся к нашей программе, добавленную функцию NXTCam размещаем в кадре инициализации программы и создаем константы для терминала **Input port** и **Input of NXTCam**. В качестве команды указываем **Enable Tracking**, запуск работы камеры.



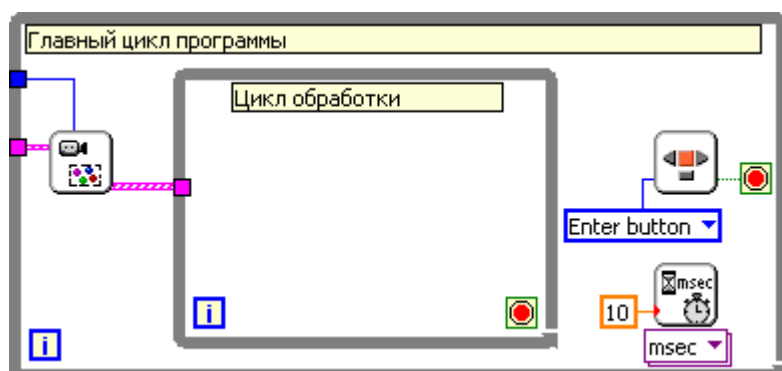
В главный цикл программы добавляем функцию чтения информации о всех объектах текущего кадра,

NXT_cam_get_all_obj , Functions -> mindsensors device -> **NXT_cam_get_all_obj**.

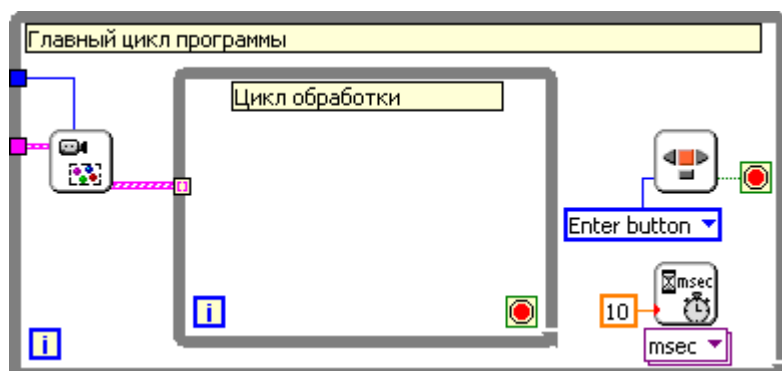
В отличие **NXTCam**, эта функция на выходе имеет не один кластер с информацией об одном из объектов, а массив кластеров. Каждый кластер массива соответствует определенному объекту, номер которого совпадает с индексом массива.



Создаем цикл поочередной обработки данных с камеры. Заводим выходной массив с данными об объектах с функции **NXT_cam_get_all_obj** через входной туннель в массив обработки. Не забываем добавить поясняющие комментарии, «главный цикл программы» и «цикл обработки».

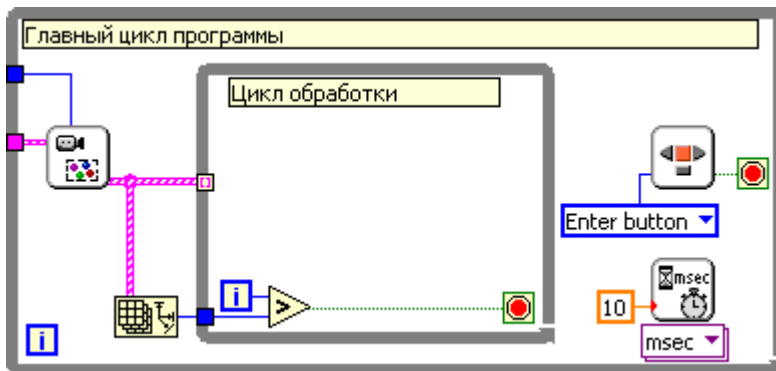


Включаем индексацию входного туннеля цикла обработки.

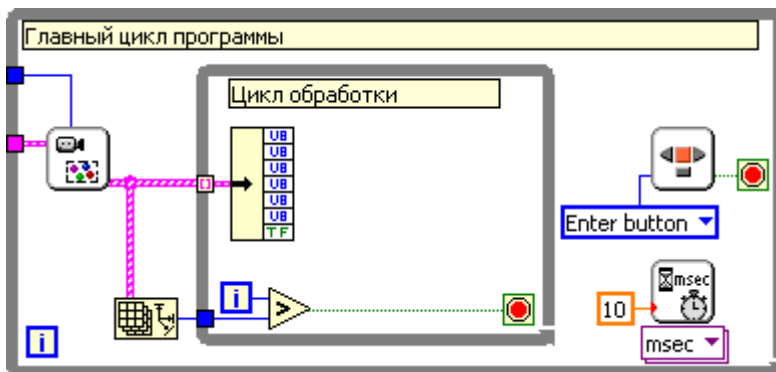


Нам нужно добавить условие завершения цикла обработки. В нашем случае он должен завершаться, когда мы обработаем все найденные в кадре объекты, то есть он должен выполняться ровно столько раз, сколько элементов находится в массиве с данными об объектах.

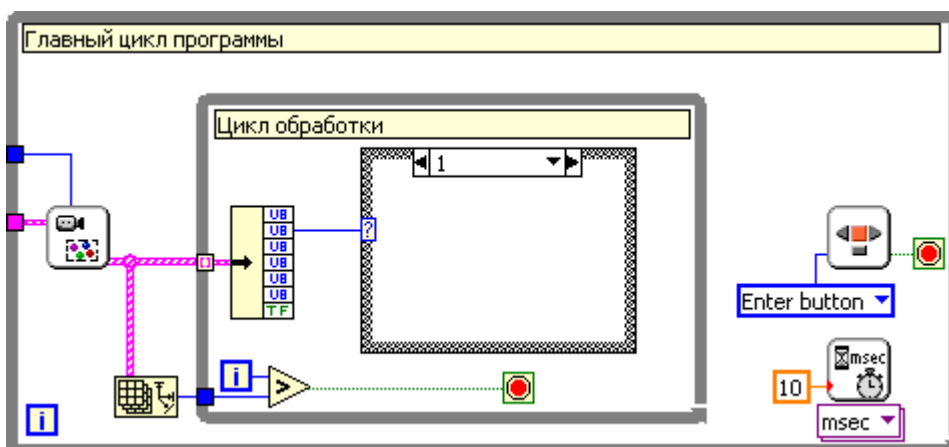
Добавляем функцию определения размера массива и используем это значение в формировании условия завершения цикла обработки.



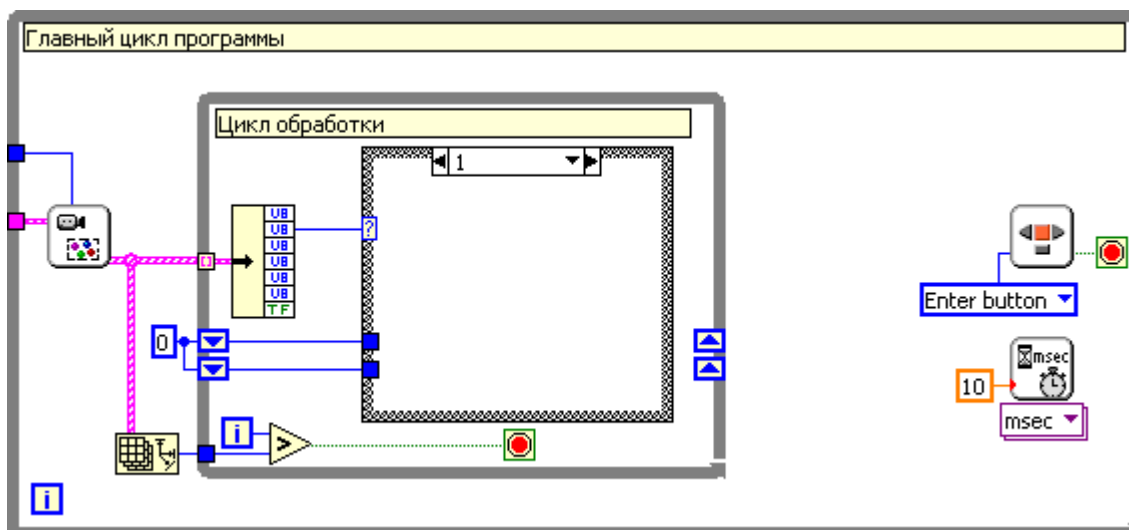
Разбираем кластер с данными о текущем объекте при помощи функции **Unbundle** , **NXT programming -> Cluster -> Unbundle**. При подключении кластера, на выходе функции автоматически появится количество терминалов соответствующее количеству элементов кластера.



Далее нужно произвести сортировку объектов по цвету. Добавляем структуру ветвления и к терминалу параметра селектора присоединяем второй (сверху), выходной терминал функции **Unbundle**, это значение номера цвета объекта.

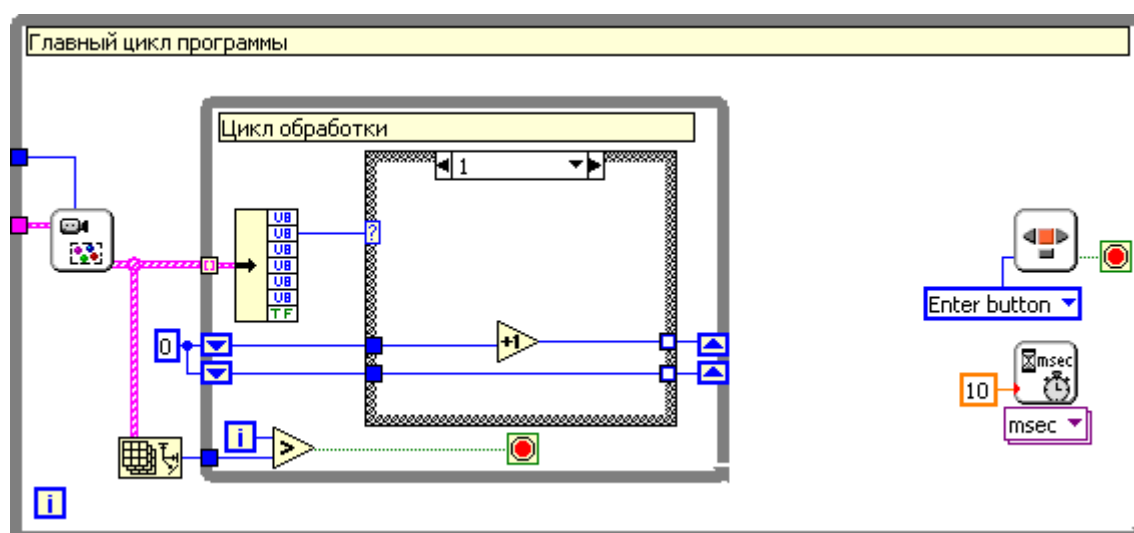


Создаем два счетчика конфет для двух различных цветов при помощи сдвиговых регистров в цикле обработки (правой кнопкой мыши на рамке цикла и в меню **Add shift register**). В качестве значения инициализации принимаем ноль, для этого создаем числовые константы через меню функций **NXT programming -> Numeric -> Numeric constant**. Текущее значение регистра заводим в ветвление по цвету объекта.

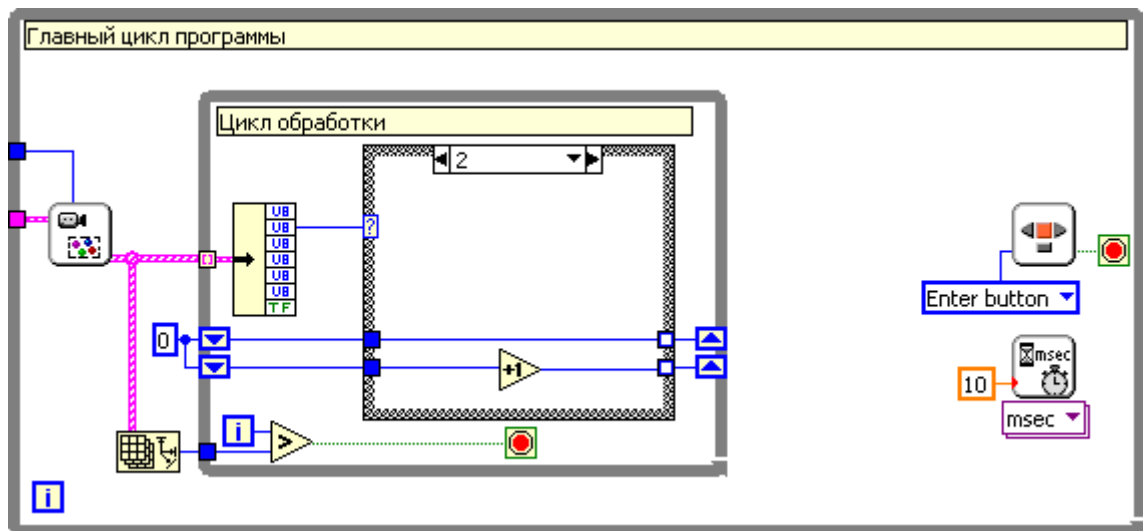


Проверяем что текущая выбранная ветка ветвления – «1», что соответствует первой ячейке из палитры цветов камеры.

Если объект имеет цвет «1», значит увеличиваем значение счетчика объектов с этим цветом, а счетчик объектов другого цвета оставляем без изменений.



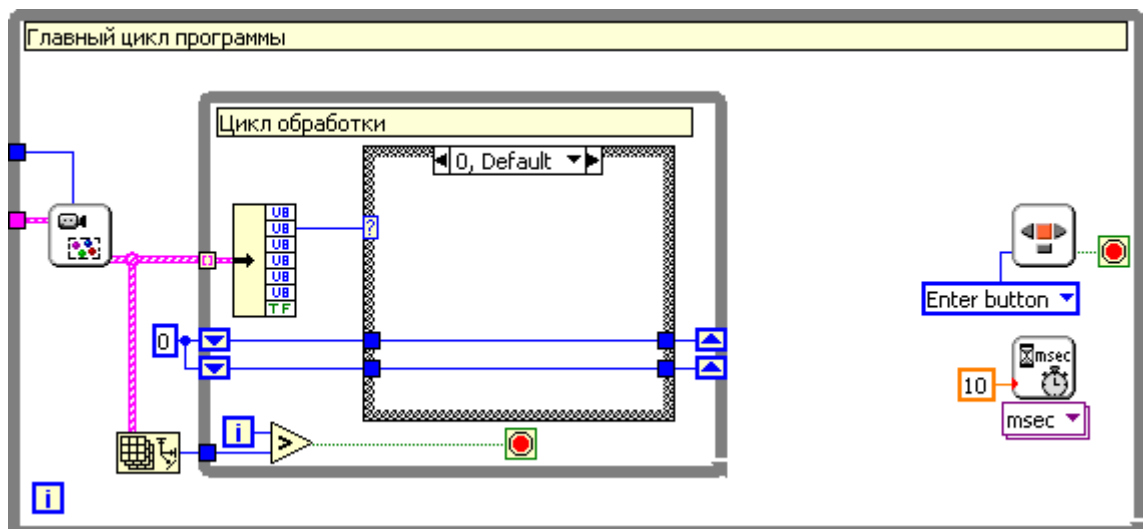
Добавляем ветку для значения 2, нажатие правой кнопкой мышки на рамке структуры ветвления и в меню -> **Add case After**. Новая ветка будет соответствовать второй ячейке из палитры цветов камеры. Проводим ту же операцию что и в ветке для «1», только для второго счетчика.



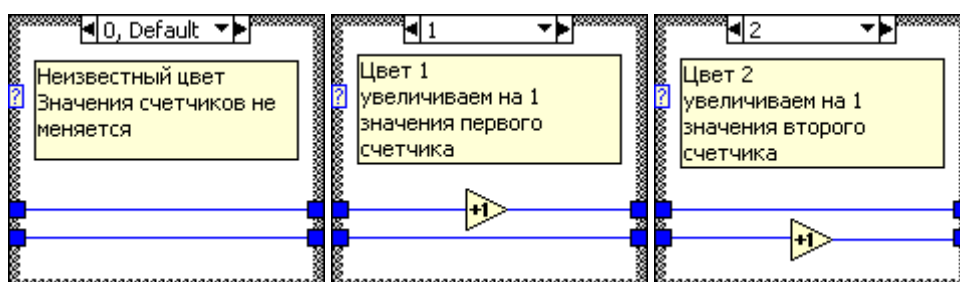
Выходные терминалы счетчиков, из структуры ветвления имеют белую заливку, это означает, что в одной из веток программа не знает какие значения выводить через эти терминалы.

При помощи клавиш переключения между ветками структуры, находим ту в которой выходные терминалы без соединений.

В нашем случае эта ветка для значений «0» и «default». Эта ветка будет выполняться, когда значение параметра селектора либо «0», либо не «1» или «2», в обоих случаях значения счетчиков остаются без изменений, просто замыкаем входные терминалы на выходные.

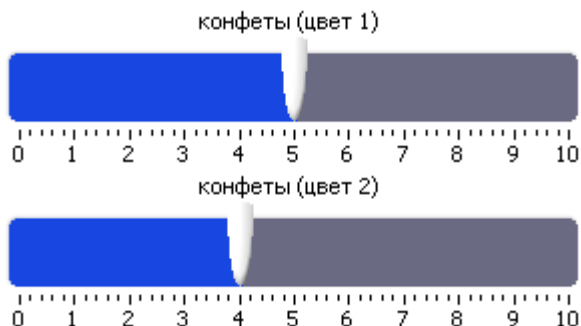


Добавляем в каждую из веток комментарии о том, при каких условиях выполняется эта ветка, и краткое описание, что в ней происходит.

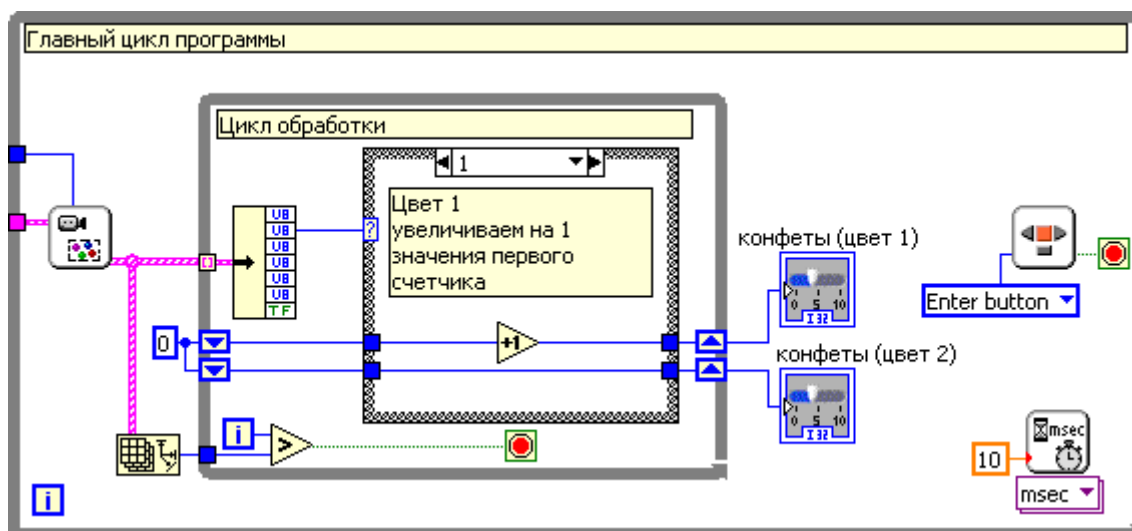


По результатам выполнения цикла сортировки на выходных терминалах сдвиговых регистров счетчиков, получаем количество в текущем кадре объектов с цветом 1, и объектов с цветом 2.

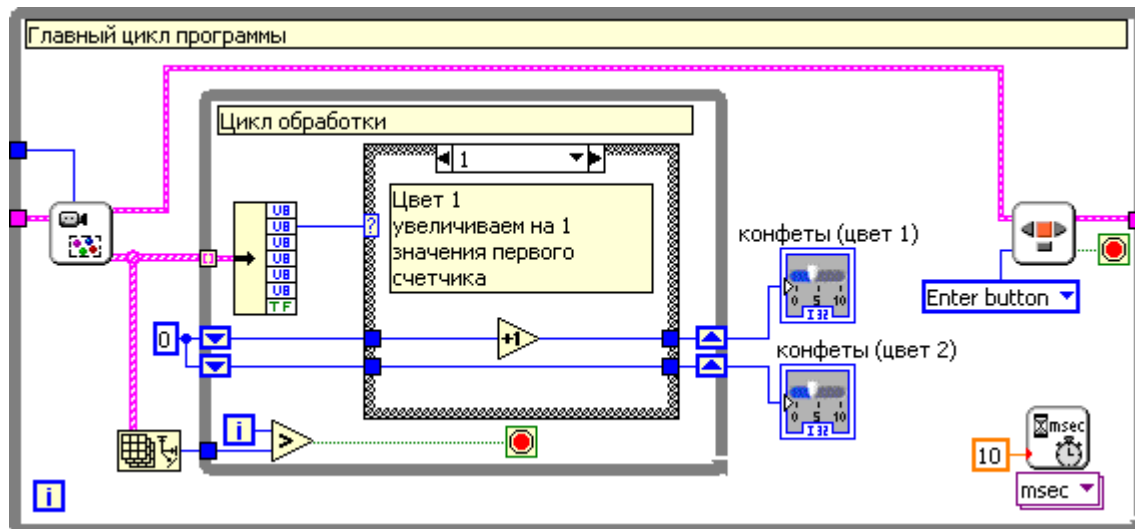
На лицевой панели создаем два числовых индикатора любого вида, например **Horizontal pointer slide**. Изначально данный вид элемента управления является контролом, для того чтобы мы могли отображать на нем значения необходимо поменять его тип на «индикатор», для этого кликаем на него правой кнопкой мыши и в меню выбираем **Change to Indicator** (Абсолютно также можно любой индикатор превратить в контрол).



Возвращаемся на панель написания диаграмм и присоединяем выходные терминалы сдвиговых регистров цикла обработки к терминалам индикаторов.



Соединяем функции **NXT_cam_get_all_obj** и **Read NXT Buttons**, и выводим кластер NXT через туннель из цикла.



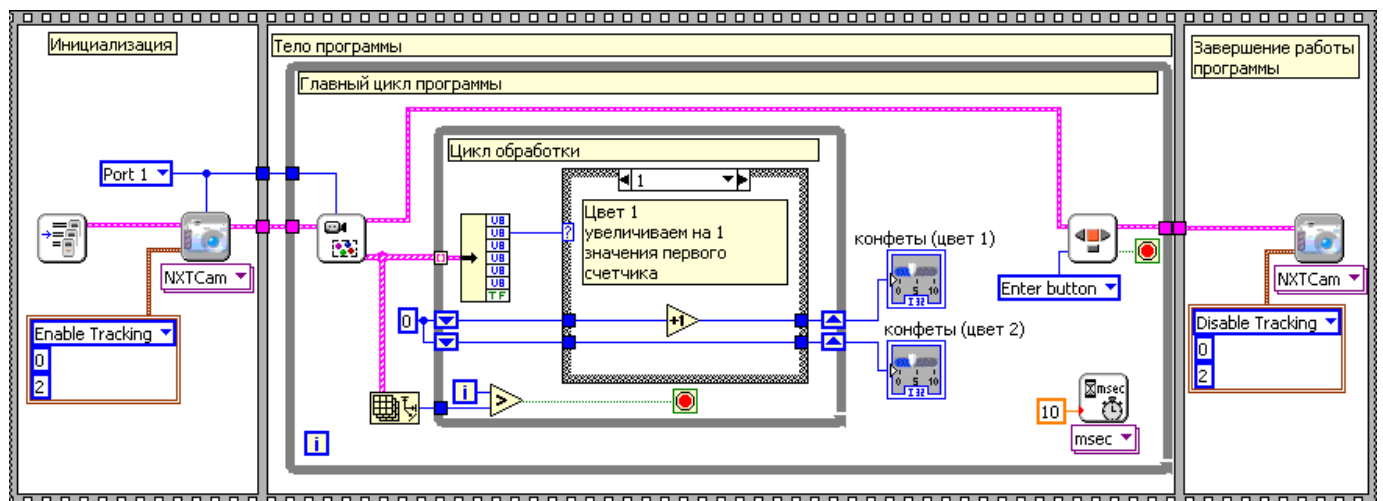
Теперь переходим к кадру завершения работы программы.

Перед завершением работы программы нам нужно корректно закончить работу с камерой, поэтому снова добавляем функцию для работы с датчиками mindsensors, **Functions -> mindsensors**



device -> mindsensors device **Acceleration Sensor**. В меню функции выбираем **Sensor -> NXTCam**. Создаем константы для терминала **Input port** и **Input of NXTCam**. В качестве команды указываем **Disable Tracking**, остановка работы камеры.

Присоединяем кластер NXT



Программа готова!

Дополнительное задание

Самостоятельно сделать индикацию количества конфет одного и другого цвета на дисплее NXT,



при помощи функций **Text**, **NXT I/O -> Display control**, и **Number to string**, **NXT programming -> String -> String/Number conversion -> Number to string**.