

# NI LabVIEW + TelegramBotAPI

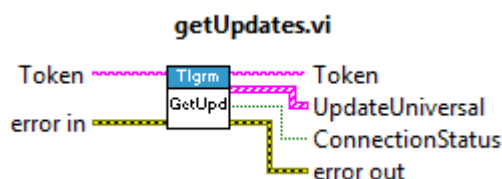
LabVIEW 2015 or later

Обратная связь: [vadimladik@gmail.com](mailto:vadimladik@gmail.com)

Эти примеры демонстрируют использование TelegramBotAPI в LabVIEW. Новичкам будет полезно для начала прочесть [Bots: An introduction for developers](#).

Для получения ботом обновлений используется метод **getUpdates** (long polling), для отправки сообщений — метод **sendMessage**. На данный момент реализованы только эти два метода. Каждое обновление представляет собой массив JSON—сериализованных объектов **Update**.

## getUpdates



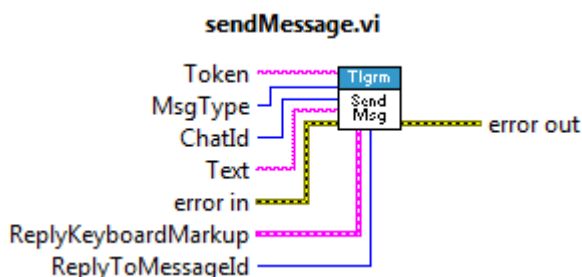
Входные терминалы

Token	token используемого бота.
error in	входной кластер ошибки.

Выходные терминалы

Token	token используемого бота.
UpdateUniresal	массив объектов Update.
ConnectionStatus	индикатор статуса соединения.
error out	выходной кластер ошибки.

## sendMessage



Входные терминалы

Token	token используемого бота.
MsgType	тип передаваемого сообщения (Text, ReplyKeyboardMarkup, ReplyKeyboardRemove).
ChatId	идентификатор активного чата.
Text	текст передаваемого сообщения.
ReplyKeyboardMarkup	объект, представляющий собой <a href="#">пользовательскую клавиатуру</a> .
ReplyToMessageId	идентификатор родительского сообщения (в групповых чатах).
error in	входной кластер ошибки.

Выходные терминалы

Error out	выходной кластер ошибки.
-----------	--------------------------

## Примеры ботов.

Для корректной работы каждого примера:

1. Необходимо [создать бота в Telegram](#) и вписать в поле "Token" токен, [выданный @Botfather](#). Токен должен начинаться с "bot", например "[bot275887199:AAHmg4xfDwkElXKcgoITf0nrYVpOVITc8k0](#)"
  2. На своём устройстве начать личный чат с ботом, который был создан на предыдущем шаге.
- 2.1 Для тестирования можно использовать бота [@ATISIndBot](#)

## TelegramBotSimpleExample

Рассмотрим пример простейшего бота в LabVIEW. Для этого необходимо открыть [TelegramBotSimpleExample.vi](#). На рисунке 1 представлена его лицевая панель, которая проста и не требует пояснений, на рисунке 2 — блок диаграмма.

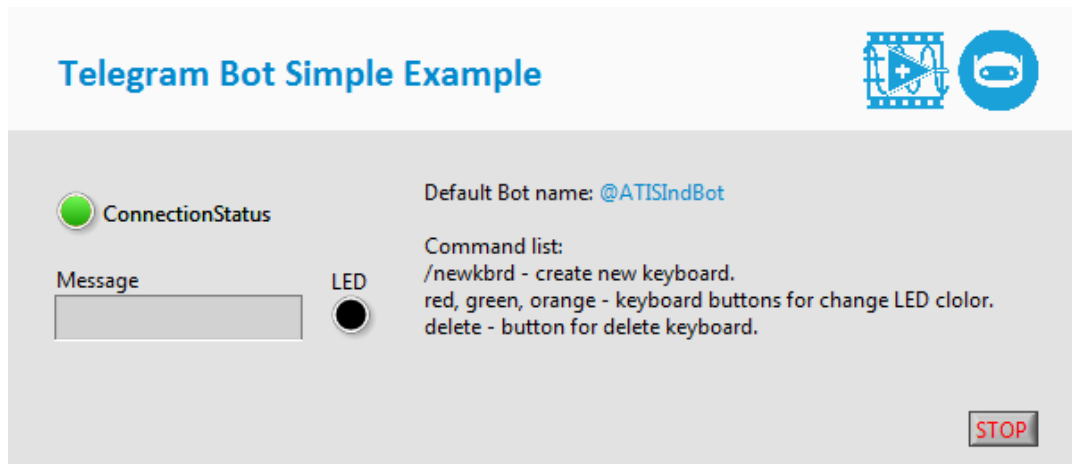


Рисунок 1 – Лицевая панель TelegramBotSimpleExample.vi

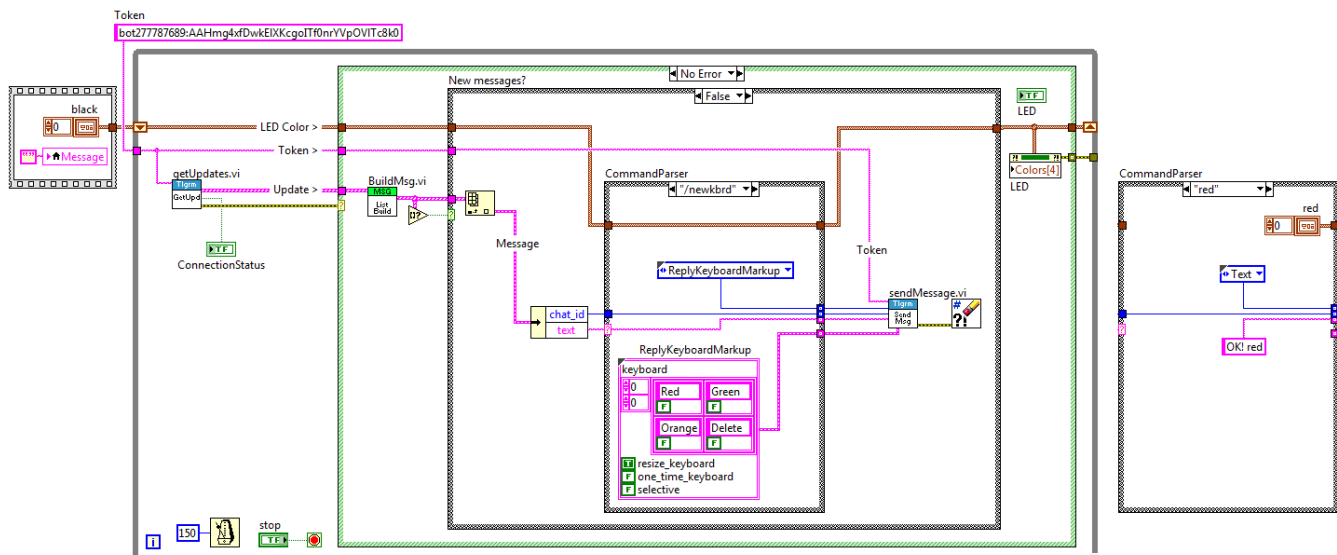


Рисунок 2 — Блок диаграмма TelegramBotSimpleExample.vi

Алгоритм работы бота очень прост. Сначала проходит инициализация. Далее в бесконечном цикле с заданным периодом происходит опрос серверов Telegram на наличие новых обновлений. SubVI *BuildMsg.vi* преобразует JSON объекты **Update** в сообщения. При наличии новых сообщений происходит сопоставление их текста вариантам Case-структуры. На последнем этапе выполняются отправка ответного сообщения в активный чат.

## Создание пользовательской клавиатуры.

Для передачи в чат клавиатуры необходимо выполнить два действия:

1. Указать тип сообщения для передачи как **ReplyKeyboardMarkup**.
2. Заполнить одноименный кластер.

Чтобы удалить клавиатуру необходимо указать тип сообщения как **ReplyKeyboardRemove**.

На следующем рисунке показан пример создания пользовательской клавиатуры, которая состоит из четырёх кнопок (Red, Green, Orange, Delete).

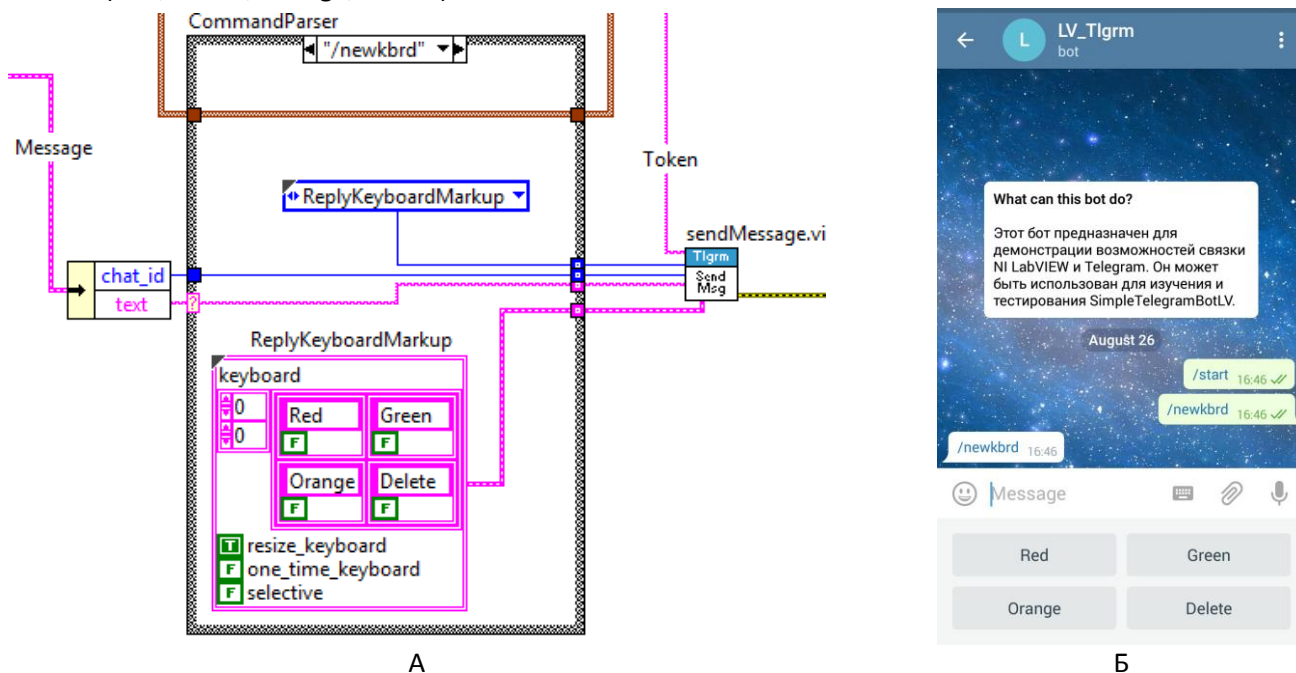


Рисунок 3 — Пример создания пользовательской клавиатуры:

А — фрагмент блок диаграммы, отвечающий за создание и отправку клавиатуры.

Б — результат, отображаемый в чате.

## myRIO+TelegramBOT Demo

Это более сложный пример реализации логики работы бота, хотя смысл его работы остаётся прежним. В данном примере с помощью сообщений в мессенджере Telegram, можно изменять характер свечения пользовательских светодиодов LED1—3 на myRIO 1900 и выполнять некоторые другие функции.

Команды, которые используются для изменения характера свечения LED:

**/led#,Period(ms/us),DutyCycle(%)** — изменяет характер свечения User LED-а (1-3). Например, **/led1,1000,50**. Для LED1, 2 период задаётся в ms, для LED3 в us.

**/led#** (без параметров), где # - номер User LED-а (1-3). Например, **/led1**. В результате бот вернёт пользовательскую клавиатуру с несколькими кнопками, позволяющими зажигать/тушить светодиод.

Команда, используемая для получения данных о погоде в определённом городе:

**/getweather,City,Region**, где City — обязательный параметр, Region — необязательный параметр.

Например **/getweather,miami,us** — бот вернёт сообщение с данными о погоде в Майами, США.

Команда, используемая для получения текущего времени myRIO:

**/gettime** — бот вернёт сообщение о текущем времени myRIO.