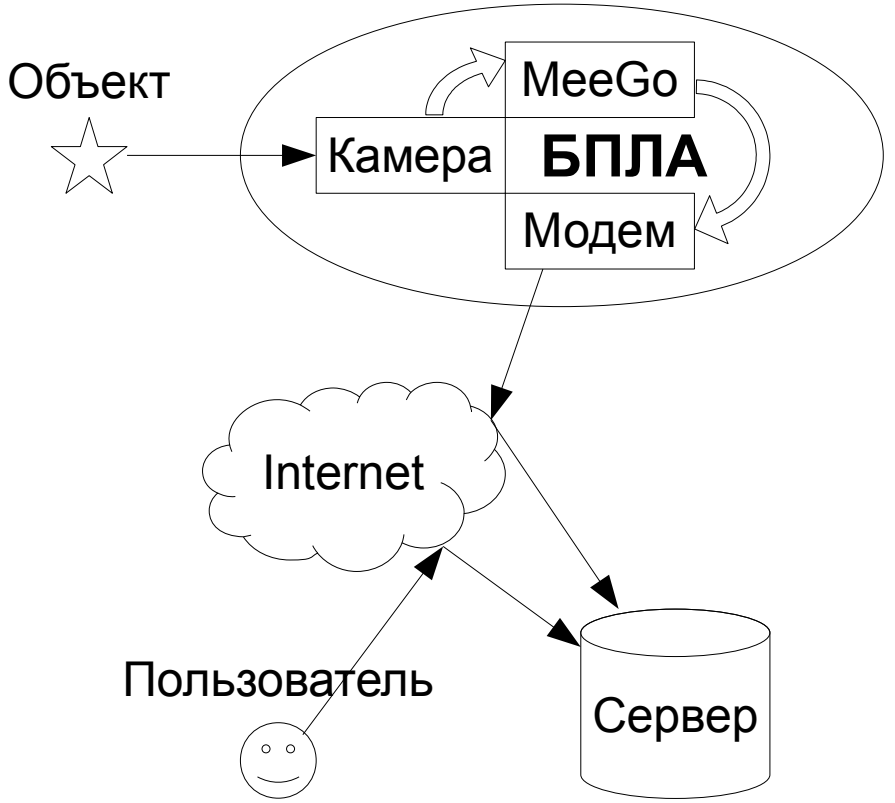
Лабораторная работа №9

Беспилотный летательный аппарат: Захват видеоизображения

Постановка общей задачи

- **Получить кадр от видеокамеры**
- Отправить кадр на сервер через GSM USB-модем
- Сохранить кадр на сервере

  и обеспечить к нему доступ пользователей



Захват кадра с камеры – варианты

- **Использование интерфейса v4l2, поддерживаемого ядром MeeGo**
- Специальные утилиты, такие как luvcview
- Библиотека gStreamer
- Библиотека QT Phonon поверх gStreamer, штатно отсутствует в MeeGo
- QT Mobility/Multimedia, в будущем должна заменить Phonon

Сборка и использование v4l2grab

- Уже есть готовая программа!

  http://www.twam.info/linux/v4l2grab-grabbing-jpegs-from-v4l2-devices
- Устанавливаем зависимости

  $> *yum install gcc libjpeg-devel*
- Собираем прямо в MeeGo

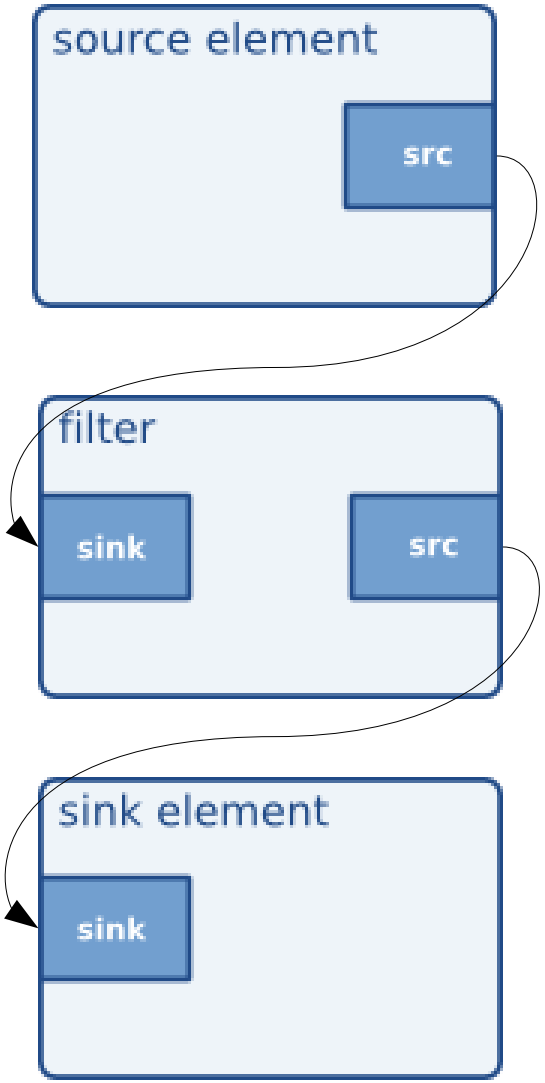
  $> *gcc v4l2grab.c -o v4l2grab -ljpeg*
- Подключаем камеру и проверяем

  $> *./v4l2grab -d /dev/video0 -o image.jpg*

Давайте попробуем другой вариант

- Использование интерфейса v4l2, поддерживаемого ядром MeeGo
- Специальные утилиты, такие как luvcview
- **Библиотека gStreamer**
- Библиотека QT Phonon поверх gStreamer, штатно отсутствует в MeeGo
- QT Mobility/Multimedia, в будущем должна заменить Phonon

Идея gStreamer – конвейер обработчиков медиапотоков



Источник — видеокамера

Фильтр = приемник+источник
Конвертер или кодек

Приемник — экран или файл

Создаем основные элементы

```
GstElement *pipeline = gst_pipeline_new("test");

// Источник – устройство v4l2
GstElement *camera_src =
        gst_element_factory_make("v4l2src", "camera_src");

// Приемник – фиктивный, но с обработчиком
GstElement *sink =
        gst_element_factory_make("fakesink", "capture_sink");
g_object_set(G_OBJECT(sink), "signal-handoffs", TRUE, NULL);
g_signal_connect(G_OBJECT(sink), "handoff",
                        G_CALLBACK(capture_callback), NULL);
```

Конвертируем видеопоток

```
GstElement *csp_filter =
gst_element_factory_make("ffmpegcolorspace", "csp_filter")

// Можно и не задавать, но мы хотим знать параметры
GstCaps *caps = gst_caps_new_simple("video/x-raw-rgb",
                "width", G_TYPE_INT, 640,
                "height", G_TYPE_INT, 480,
                "bpp", G_TYPE_INT, 24,
                "depth", G_TYPE_INT, 24,
                "framerate", GST_TYPE_FRACTION, 15, 1,
                NULL);
```

Собираем конвейер и запускаем

```
gst_bin_add_many(GST_BIN(pipeline), camera_src, csp_filter, sink, NULL);

gst_element_link_many(camera_src, csp_filter, NULL);
gst_element_link_filtered(csp_filter, sink, caps);

// Указываем начать обработку данных
gst_element_set_state(GST_ELEMENT(pipeline), GST_STATE_PLAYING);

// Теперь управление берут на себя gStreamer и его события
g_main_loop_run(loop);
```

Обработчик для захвата кадра

```
gboolean capture_callback(GstElement *element,
                          GstBuffer *buffer, GstPad *pad, void *data){

    // Буфер передается без заголовков, даже без длины!
    if(!create_jpeg("image.jpg",GST_BUFFER_DATA(buffer)))
        g_error("create_jpeg() fail!");

    // После захвата завершаем gStreamer
    g_main_loop_quit(loop);
    return TRUE;
}
```

Сохранение в JPEG

- create_jpeg() — типичное использование libjpeg для сжатия буфера
- В буфере нет никаких параметров, все настройки сжатия должны соответствовать настройкам фильтра!

Сборка и использование приложения gStreamer

- Устанавливаем зависимости

  $> *yum install gcc libjpeg-devel gstreamer-jpeg gstreamer-ffmpeg*

- Собираем прямо в MeeGo

  $> *gcc gstgrab.c -o gstgrab -ljpeg `pkg-config --cflags --libs gtk+-2.0 gstreamer-interfaces-0.10`*

- Подключаем камеру и проверяем результат в image.jpg

  $> *./gstgrab*

Сравнение v4l2grab и gStreamer

v4l2grab

- Размер кода 30 Кб
- Время разработки — 30 минут
- Больше реализованных возможностей
- Зависит только от ядра Linux и libjpeg

gStreamer

- Размер кода 3 Кб
- Время разработки — 6 часов
- Больше потенциальных возможностей
- Больше зависимостей
- Лучше переносимость вне Linux

Домашние задания

1. Интегрировать v4l2grab в QT-приложение
2. Создать QT-обертку для v4l2grab
3. Научиться захватывать не первый доступный кадр, а десятый в v4l2grab и gStreamer

Вопросы?

avkoryavko@gmail.com