

ПОДХОД К СОЕДИНЕНИЮ МНОЖЕСТВА КАМЕР С RASPBERRY PI ДЛЯ СЪЕМКИ СФЕРИЧЕСКИХ ПАНОРАМ

Стракович А.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Луцик Ю.А. – к.т.н, доцент

Показан подход к получению множества изображений для создания сферической панорамы, основанный на использовании множества камер. Рассмотрены проблемы которые могут возникнуть при подключении камер и способы их решения.

Сферическая панорама представляет из себя собранное из множества отдельных кадров изображение в сферической проекции. Характерной чертой сферических панорам является максимально возможный угол обзора пространства (360×180 градусов). Просматривая такие панорамы создается эффект присутствия на рассматриваемом месте.

Для создания сферических панорам, необходимо получить множество фотографий одновременно либо за минимальное время, чтобы движущиеся объекты находились на одном и том же месте на разных фото. Иначе совместить такие фото будет невозможно, либо на итоговой панораме случайно попавший в кадр движущийся объект будет прозрачным.

Основным управляющим устройством, для работы с камерами, может выступать практически

любой микроконтроллер с GPIO. В представленном подходе используется плата Raspberry PI 3B+[1]. Используемые камеры OV2640[2]. Для работы с камерой необходим тактирующий сигнал с частотой 24 МГц или больше. Он используется для синхронизации выдачи камерой очередного байта изображения и принятия данного байта Raspberry. Механизм синхронизации можно реализовать несколькими способами.

Первый подход – генерация сигнала самой Raspberry. Программной генерацией, в зависимости от используемых библиотек, можно получить частоту от 15 кГц до 22 МГц[3]. Недостатком данного подхода является то, что он медленный и нестабильный, с точки зрения получаемой частоты.

Второй подход – использование внешнего тактового генератора. Такой генератор всегда будет выдавать стабильный сигнал необходимой частоты. Необходимо только согласовать напряжение генератора и Raspberry, если генератор формирует сигнал отличный от требуемого для Raspberry PI.

У Raspberry PI 28 GPIO пинов. Для подключения камеры нужны 14 пинов. Из них 12 для получения изображения и 2 для инициализации камеры по интерфейсу SCCB (совместим с I2C). В таком случае получится подключить только 2 камеры. Возникает проблема нехватки пинов для подключения всех камер одновременно. В качестве согласующего элемента предусмотрен блок коммутации, который в определенный момент времени передает данные изображения только одной из камер. Реализацию блока коммутации можно выполнить несколькими способами.

Первый способ заключается в реализации отдельной схемы, что представляется аппаратно более избыточным.

Второй способ предполагает использование ПЛИС, на которой реализовать схему из первого способа. Данный подход является более гибким, т.к. при любом изменении конфигурации коммутирующего устройства не требуется изготавливать новую схему, а только загрузить конфигурацию в ПЛИС. Для реализации этого подхода используется ПЛИС EPM240T100C5[4].

Общая схема представлена на рисунке 1. К выбранной ПЛИС можно подключить 6 камер. При использовании камер с объективом типа “рыбий глаз” их хватит для полного покрытия обзора 360x180 градусов, что позволит сделать сферическую панораму.

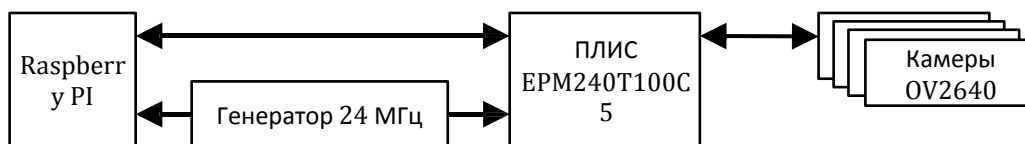


Рисунок 1 – Схема подключения

Список использованных источников:

1. Raspberry PI 3B+ Datasheet [Электронный ресурс] – Режим доступа : www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf – Дата доступа : 01.03.2020.
2. Камера OV2640 Datasheet [Электронный ресурс] – Режим доступа : www.uctronics.com/download/cam_module/OV2640DS.pdf – Дата доступа : 01.03.2020.
3. Benchmarking Raspberry Pi GPIO Speed [Электронный ресурс] – Режим доступа : www.codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/ – Дата доступа : 10.03.2020.
4. EPM240T100C5 Datasheet [Электронный ресурс] – Режим доступа : www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max2/max2_mii5v1.pdf – Дата доступа : 12.03.2020