

# Встроенные видеосистемы, использующие AVR-8-битный микроконтроллер Atmega64 и сенсор Omnivision OV6620

Автор: H.N. Nguyen and K.J. Lin

Автор перевода: И.С. Пильгук

Источник: Embedded Vision System Using AVR-8 bit Microcontroller Atmega64 and Omnivision OV6620 CMOS Image Sensor

Аннотация - в этом проекте, мы рассматриваем небольшой, с низким энергопотреблением, работающий в режиме реального времени процессор обработки изображения, позволяющий отслеживать цветные объекты. Наша система использует недорогой Atmel AVR 8-разрядный микроконтроллер ATmega64, который был подключен непосредственно к модулю камеры C3088 с помощью датчика изображения Omnivision OV6620 CMOS. Это означает, что на этой скромной аппаратной архитектуре, мы будем выполнять простые алгоритмы компьютерной обработки изображений, ранее нуждающиеся в значительном количестве вычислительных ресурсов, потребляющих много энергии, в месте, недоступном для такого небольшого, дешевого оборудования. Он был специально разработан как недорогая, с низким энергопотреблением, но эффективная встроенная видеосистема с простым оборудованием, которое может быть легко добавлено в другое приложение (робот, безопасность, мониторинг и т.д.). Особое внимание к программному алгоритму, эффективность быстрого отслеживания цветных объектов со скоростью 30 кадров / сек осуществляется за счет аппаратных средств. Это функциональная видеосистема, с четко определенным интерфейсом, который доступен через стандартный последовательный порт, обеспечивая высокий уровень, после обработки графической информации в первичной системе (PC, другой микроконтроллер и т.д.). Это избавляет вас от необходимости выполнения обработки изображений на главной системе, и позволяет разработчику сосредоточиться на том, как использовать на высоком уровне, данные изображения для выполнения поставленной задачи.

## I. ВВЕДЕНИЕ

Роботизированные системы становятся более компактными, потребляют меньше энергии, дешевле, что позволяет применять их в областях, которые ранее были недоступными, и это также верно для системы технического зрения. Есть много относительно простых компьютерных алгоритмов, которые были чрезвычайно полезными в различных приложениях, [1], [4], [6], [7]. Тем не менее, оборудование для их реализации, как правило, сложное и дорогостоящее. Традиционно эти системы состоят из камеры, устройства захвата кадров, а также мощного компьютера для связи захвата кадров и выполнения алгоритмов. С появлением недорогих цветных CMOS камер и с высокоскоростных микроконтроллеров, стало возможным создавать простые и дешевые системы. [2] Известная CMUCAM2 которая использует камеру Omnivision OV6620 CMOS, микроконтроллер Uvicom SX52, работающий на 75 МГц, чип буфера кадра AL422B - это значительный шаг вперед, простой и дешевый. Однако,

в некоторых небольших приложениях, таких как движущиеся роботы, слежение за объектом ... это все еще довольно сложный и дорогой продукт с ценой 199 USD.

В технологии камер CMOS, схема захвата пикселей интегрирована с аналого-цифровым преобразователем, так что нет необходимости иметь отдельный захватчик кадров. Микроконтроллеры становятся все более мощными, быстрыми и их стоимость продолжает снижаться. Тем более, Atmel семейства AVR 8-разрядный микроконтроллер может работать на максимальной скорости 20 МГц со стоимостью всего несколько долларов. Такой микроконтроллер очень потребляем для встроенных видеосистем, даже по сравнению с CMUCAM2, который использует 75 МГц Uvicom SX52 микроконтроллер. Тем не менее, мы с особым вниманием рассмотрели алгоритм эффективного функционирования встроенной видеосистемы, который может быть построен с использованием оборудования захвата изображения и процессора. Мы построили в режиме реального времени процессор обработки изображений, позволяющий отслеживать цветные объекты, который состоит из Omnivision OV6620 CMOS камеры, 2 Atmel AVR-8bit микроконтроллеров (Atmega64 работает на 16 МГц в качестве основного микроконтроллера ATmega8535 - в качестве загрузочного процессора) со стоимостью менее 100 долларов США. Быстрая и дешевая сегментация цветных изображений была реализована для достижения результата. Мы представляем всю структуру в этой статье.

## II. Архитектура системы

Наша система предназначена для обеспечения частоты кадров цветного видео, с низкими издержками хост-интерфейса для использования в качестве простой системы технического зрения, а также базового захвата видео. Информация из захваченного изображения передается в основную систему (PC, другой встроенный контроллер и т.д.) для использования. В обычном приложении внешнего процессора в мобильном интерактивном роботе можно настроить цвет слежения видеосистемы в режиме обратного отслеживания пакетов. Основной микропроцессор затем обрабатывает данные в режиме реального времени и выводит на высоком уровне после обработки информации. [5]

### A. Аппаратная часть

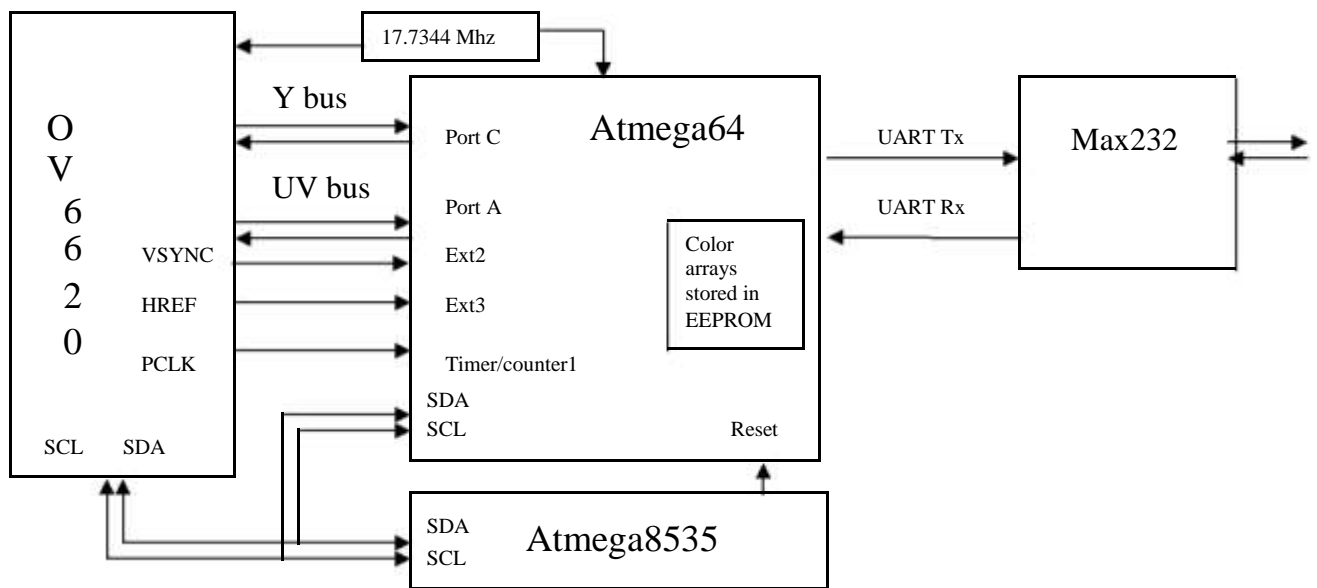


Рис. 1. Схема оборудования.

Аппаратная нашего проекта очень проста. Она состоит из четырех чипов: Omnivision OV6620 CMOS, Atmega64 в качестве основного процессора, ATmega8535 как "загрузочный контроллера I2C обеспечивает связь для интеграции в другой встроенный контроллер и простой сдвиг уровня для RS-232 последовательных данных. Для того чтобы упростить дизайн, Omnivision OV6620 КМОП-датчик подключается непосредственно к мега64 для получения доступа к важным сигналам камеры (пиксельный таймер горизонтальной и вертикальной синхронизации, и, конечно же, шина данных). Мега64 ждет входящий поток данных с камеры и обрабатывает их в реальном времени. Затем он отправляет извлеченную обработанную информацию высокого уровня во внешний мир через внутренний порт UART.

Ввод изображений в системе обеспечивается CMOS камерой на чипе Omnivision OV6620 [3]. Камера CMOS установлена на несущей плате, которая включает объектив и поддержку пассивных компонентов. Сама по себе плата свободно запускается и выводит на поток 8-битные RGB или пиксельные YCrCb цвета. Синхронизированные сигналы, в том числе пиксельные таймеры, используются для считывания данных и указывают новые кадры и горизонтальные линии. OV6620 поддерживает разрешение до 352 x 288 с максимальной частотой обновления 360 кадров в секунду (FPS) и параметры камеры, такие как насыщенность цвета, яркость, контрастность, баланс белого, время экспозиции, усиления и выходных режимов программируется с помощью стандартного последовательного интерфейса I2C. Аналоговый монохромный выход существует, что может быть использовано для мониторинга внешнего объекта.

Основной микроконтроллер использует для обработки видеоданных RISC-архитектуру микроконтроллера ATmega64, работающего на 17,7344 МГц. При выполнении большинства инструкций за один такт, ATmega64 достигает производительности

приблизительно 1 MIPS на МГц, что позволяет разработчикам систем оптимизировать энергопотребление по сравнению со скоростью обработки данных. [9] Это обеспечивается следующими возможностями: 64К байт-программируемой флэш с Read-While-Write, 2К байт EEPROM, 4 Кбайт SRAM, 53 общего назначения линий ввода / вывода, 32 общего назначения рабочих регистров, счетчик реального времени (RTC), четыре гибких таймера / счетчика с режимами сравнения и ШИМ, два УСАПП (USART0 будет использоваться для связи с ПК), байт-ориентированный двухпроводной последовательный интерфейс, который будет использоваться в качестве интерфейса I2C с модулем камеры, 8-канальный 10-разрядный АЦП с опциональным дифференциальным входным каскадом с программируемым коэффициентом усиления, программируемый сторожевой таймер с внутренним генератором, SPI последовательный порт, IEEE Std. 1149,1 совместимый интерфейс JTAG тест, который также используется для доступа к системе отладки и программирования, и шесть программ режима энергосбережения. В режиме ожидания останавливается процессор, позволяя SRAM, таймерам / счетчикам, SPI порту и системе прерываний продолжать функционировать. Эта функция помогает нам сократить потребление энергии, потому что это выгодно завершить работу системы, когда она не нужна.

Отдельный чип контроллера загрузки необходим, потому что основной чип процессора камеры (Мега64) тактируется частотой 17.7344MHz - сигнал из видео модуля, который не доступен при включении питания. При включении питания, "загрузка процессора" (Мега8535) содержит основной процессор в режиме сброса, в то время как он выдает команды настройки для видео модуля. Высокая частота включается для выхода из управления основным процессором. Загрузка процессора выдает сигналы управления камерой и позволяет центральному процессору продолжить работу.

## В. Прошивка

Все прошивки были написаны на C и компилируются с помощью компилятора WinAVR. При компиляции текущей прошивки требует 13 Кбайт FLASH ROM и используется 768 байт EEPROM.

**Обработка данных "на лету".** Прошивка играет важную роль в успехе нашего проекта. ATmega64 ограничивает ОЗУ, ПЗУ. Мы не используем буфер кадра так как программирование было бы ограничено сроками, требования обусловлены обработкой данных "на лету". Это означает, что обработка должна быть сделана с потоком данных, поступающим от камеры. Основной цикл обработки будет выглядеть следующим образом: Пуск цикла; Подождите, действительно пиксель на шину данных (PCLK высокий); Прочитайте значение красного от шины данных Y, прочитайте значение зеленого от УФ шины данных; Подождите, синий значение по шине данных (PCLK низкого и высокого раз); образец синий значение УФ шины данных, выполнять обработку изображений; конец цикла. Все эти обработки должны быть сделаны примерно за 16 тактов с полной скоростью потока изображения от OV6620. Тем не менее, мы можем тратить 3 или 4 цикла для мониторинга, когда PCLK изменяет состояние, а цикл назад, чтобы проверяться еще раз, если он не изменил состояние. Чтобы решить эту проблему, мы используем тот же источник синхронизации для mega64 и модуль камеры, чтобы мы могли быть уверены, что каждые 6 тактов, данные по шине данных будут изменять значение, и будут готовы для чтения. Таким образом, мы не должны проверять линии PCLK между пиксельных данных, просто синхронизировать один раз с ним в начале каждой строки, устранять все эти циклы проверки состояния PCLK.

**Цветовое отслеживание объекта.** [4] У нас работают быстрый и простой алгоритм цветовой сегментации, чтобы наша система способна была отслеживать до 8 цветных объектов на 30 Гц [1]. Этот подход предполагает использование пороговых значений в трехмерном цветовом пространстве RGB. Чтобы проиллюстрировать подход, рассмотрим следующий пример. Предположим, что мы дискретизировали пространство RGB для 10 уровней в каждом измерении. Таким образом, "оранжевый", например, можно представить, назначив следующие значения для элементов каждого массива:

```
RClass[] = {0,1,1,1,1,1,1,1,1,1};
GClass[] = {0,0,0,0,0,0,0,1,1,1};
BClass[] = {0,0,0,0,0,0,0,1,1,1};
```

Таким образом, для проверки пикселей с цветовыми значения (1,8,9) является ли они членом цвеевогот класса «оранжевый» все, что нужно сделать, это вычислить выражение RClass [1] И GClass [8] И BClass [9], которое в данном случае будет преобразовано в 1, что означает истинный цвет в классе «оранжевый». Преимуществом такого подхода является то, что он может определять членство пикселя в нескольких классах одновременно. В качестве примера, предположим, что область цветового пространства занимают «синие» пиксели, представленные следующим образом:

```
RClass[] = {0,1,1,1,1,1,1,1,1,1};
GClass[] = {1,1,1,0,0,0,0,0,0,0};
BClass[] = {0,0,0,1,1,1,0,0,0,0};
```

Мы могли бы объединить "оранжевый" и "синий" представления следующим образом:

```
RClass[] = {00,11,11,11,11,11,11,11,11,11};
GClass[] = {01,01,01,00,00,00,00,10,10,10};
BClass[] = {00,00,00,01,01,01,00,10,10,10};
```

Первый (высокого порядка) бит в каждом элементе используется для обозначения "оранжевый", а второй бит используется для обозначения "синий". Таким образом, можно проверить (1,8,9) в одном из двух классов по оценке одного выражения RClass [1] И GClass [8] И BClass [9]. В результате 10, с указанием цвета в «оранжевом» классе, но не в "синем". В нашей системе каждый массив имеет 256 элементов, и каждый элемент является 8-битным числом. Таким образом, можно оценить членство 8 цветных объектов сразу с двумя операциями. Для этого требуется 8 бит разрешения для каждого цветового канала и 256x3 байт EEPROM для хранения массивов 3 цветовых каналов.

**Связанные области.** [4] После того, как различные цвета пикселей были классифицированы, связанные регионы формируются путем изучения секретных пикселей. Во многих роботах, приложениях обнаружение значительно изменяющихся соседних пикселей изображения относительно редко. Аналогично группировка соседних пикселей, одним «запуском» вычисляем кодированную длину серии (RLE). Процедура слияния сканирует соседние строки, столбцы и отслеживает те, которые того же класса, цвета и перекрываются в четырех областях. Каждый набор работает, на указанной связанной области.

## С. Интерфейс:

Как отмечалось выше, основной процессор связан непосредственно с модулем камеры. Он получает каждый пиксель блока (R, G, B), осуществляет отображение в реальный цвет, вычисляет RLE, и хранит информацию в буфере. Нет необходимости опрашивать камеру, потому что этот интерфейс синхронизируется с пиксельными данными (OV6620 и mega64 используют тот же таймер). Вот пиксели блока:

```
...G G G G... (ряд x)
...B R B R... (ряд x+1)
```

[5] Пиксель блока определяется как непрерывная группа из 4 пикселей, которые объединяются вместе, чтобы сформировать определенный цвет. Как правило, формируется выборка значения зеленого, а затем красные и синие значения (поскольку мы имеем дело с Bayer данными о цвете). Пиксельные блоки отбирают красный, зеленый и синий цвета и используются в качестве индекса в соответствующих массивах цвета. Цвет массивов возвращает значения, которые могут быть логически объединены, так что определенный триплет RGB установит один бит после логической операции «И». Этот один бит указывает, какой цвет будет установлен после логической операции. Также возможно бит не будет установлен после этой операции, указывая, что триплет RGB не соответствует ни одному из цветов в массивах. Наша система может обмениваться данными с ПК через стандартный 9-контактный последовательный порт. Мы можем управлять камерой с полным разрешением (176 x 144 пикселей) и анализировать их, чтобы определить цвета, которые присутствуют в захваченном изображении. Затем мы строим цветовые

массивы, чтобы указать, какие цвета, система должна отслеживать выбирая цвета в изображении. После того как все цвета были выбраны, цвет массива будет отправлен вниз к системе через последовательный порт.

#### D. Результаты

Наша конечная встроенная видеосистема работает с максимальной скоростью 30 кадров в секунду с разрешением до 88x144 пикселей, обеспечивая в реальном времени отслеживание статистики (количество объектов, цвет, ограничивающий прямоугольник и др.) через стандартный последовательный порт (UART). Потребляемая мощность невелика, 100 мА, 5В на полной скорости. Кроме того, система может делать цветные снимки (176x144) и отображать исходные данные Bayer или интерполировать данные. Результаты каждого отслеживаемого объекта отображаются в реальном времени с цветом и ограничивающим информацией прямоугольником.

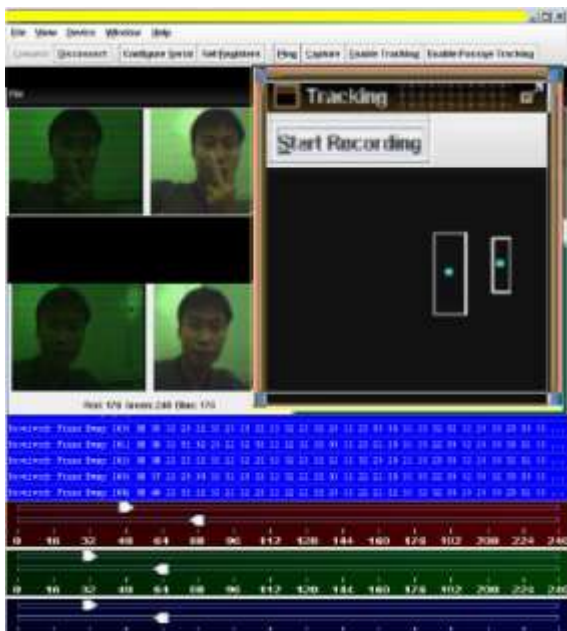


Рис. 2 Захват изображений и отслеживания объектов.



Рис. 3 Встроенная аппаратная видеосистема.

### III. ЗАКЛЮЧЕНИЕ

Целью данной работы является реализация простой встроенной видеосистемы, потребляющей мало мощности, чтобы заменить довольно сложные, дорогие системы, которые находятся вне досягаемости для многих разработчиков, у которых умеренные требования к обработке изображений. Отслеживание объектов в цвете возможно с реальной скоростью 30 кадров / сек. В будущем нам нужно будет применять систему на некоторые конкретные приложения для оценки производительности системы в различных условиях освещения. Мы также хотели бы добавить больше функциональных возможностей QCIF данных, так как это самый быстрый поток, работающий в режиме реального времени, что повышает разрешение изображения и скорость обработки изображений.

#### ССЫЛКИ

- [1] A. Rowe, C. Rosenberg and I. Nourbakhsh, "A Low Cost Embedded Vision System" *The Proceedings of IROS*, 2002.
- [2] A. Rowe, C. Rosenberg and I. Nourbakhsh, "A Second Generation Low Cost Embedded Vision System" *Proceedings of the 2005 IEEE Computer Society Conference*, 2005.
- [3] Omnivision Technologies Incorporated, "OV6620 Single-Chip CMOS CIF Color Digital Camera Technical Documentation", <http://www.ovt.com/>
- [4] J. Bruce, T. Balch and M. Veloso, "Fast and Inexpensive Color Image Segmentation for InteractiveRobots," *The Proceedings of IROS 2000*, 2000.
- [5] J. R. Orlando, Website <http://www.jrobot.net>.
- [6] I. Horswill, "Polly: A vision-based artificial agent", *The Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.
- [7] R. Sargent and B. Bailey and C. Witty and A. Wright, "Dynamic Object Capture Using Fast Vision Tracking," *AI Magazine*, vol. 18, no. 1, 1997.
- [8] I. Ulrich and I. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision", *Proceedings of AAAI Conference*, pp. 866-871, 2000.
- [9] Atmel Corp., Website <http://www.Atmel.co>