

Controlador en espacio de usuario para la cámara QHY5T

Tesina de Licenciatura en Informática
UNLP

Universidad Nacional de La Plata
Facultad de Informática

Junio 2014
La Plata, Argentina

Agenda

1

El hardware

- El sensor
- El μ procesador
- El puerto de guiado

2

El firmware

- Captura
- Reconstrucción

3

El controlador

- ¿Por qué en espacio de usuario?
- Arquitectura de un vistazo
- El controlador - Detalles de la arquitectura

4

El visor

- El visor de un vistazo
- Detalles del visor

5

Resultados obtenidos

- Rendimiento
- Resultados obtenidos

El hardware de un vistazo

- Sensor CMOS de 3 Mpx color (MT9T001)
- Resolución 2048x1536 (pixel size $3.2 \times 3.2 \mu\text{m}$)
- μproc Cypress CY7C68013A (USB/I²C)
- Arquitectura Intel 8051 + EZ-USB FX2
- Puerto de guiado opto-acoplado
- Rosca para filtros y barril de 1.25"

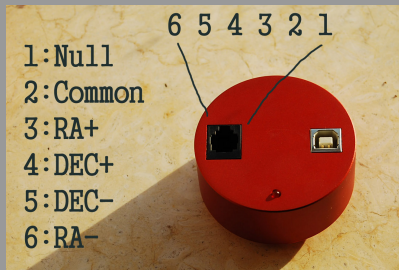


Figura: Puerto de guiado en la cámara QHY5T



Figura: Cámara QHY5T y barril adaptador de 1.25"

Detalles del sensor (cont)

- Soporte para ROI por hardware
- Binning 2x2, 3x3, y binnings no cuadrados
- Salteo de filas y columnas (skipping)
- Ganancia independiente para los canales R, G1, G2 y B

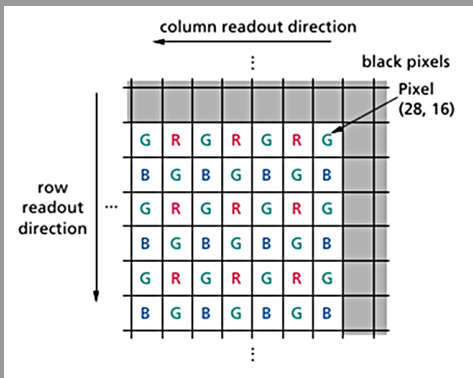


Figura: Patrón de Bayer del CFA del sensor MT9T001

Detalles del sensor (cont)

Registro	Descripción	Formato	Default
0x01	Fila de comienzo	0000 00dd dddd ddd0	0x0014
0x02	Columna de comienzo	0000 0ddd dddd ddd0	0x0020
0x03	Cantidad de filas	0000 00dd dddd ddd1	0x05FF
0x04	Cantidad de columnas	0000 0ddd dddd ddd1	0x07FF
0x05	Horizontal blanking	0000 00dd dddd dddd	0x008E
0x06	Vertical blanking	0000 00dd dddd dddd	0x0019
0x22	Modo de lectura para filas	0ddd 0ddd 0ddd 0ddd	0x0000
0x23	Modo de lectura para columnas	0000 0ddd 00dd 0ddd	0x0000
0x2B	Gain para Verde1 (G1)	0ddd dddd 0d0d dddd	0x0008
0x2C	Gain para Rojo (R)	0ddd dddd 0d0d dddd	0x0008
0x2D	Gain para Azul (B)	0ddd dddd 0d0d dddd	0x0008
0x2E	Gain para Verde2 (G2)	0ddd dddd 0d0d dddd	0x0008
0x35	Gain global	dddd dddd dddd dddd	0x0008

Tabla: Resumen de los registros más importantes y sus valores por defecto.

Detalles del μ procesador

- Reloj del sistema de 24 MHz.
- Set de instrucciones basado en la arquitectura Intel 8051, con algunas instrucciones y registros añadidos
- Cuatro Endpoints USB programables para transferencias *BULK*, *INTERRUPT* o *ISOCRONOUS*
- Transceptor USB 2.0 de alta velocidad integrado
- Soporte de compatibilidad con EZ-USB FX2
- Memoria RAM integrada de 16 KB para datos y/o instrucciones.
- Interfaz de datos de 8 o 16 bits con conversión automática.
- Interfaz I^2C (solo en modo Master) a 100/400 KHz.



La interfaz de guiado

- Conector RJ-25 (RJ-11 pero con 6 líneas)
 - ▶ Pin 1: No conectado
 - ▶ Pin 2: Conectado a Ground
 - ▶ Pin 3: Pulso en RA+
 - ▶ Pin 4: Pulso en DEC+
 - ▶ Pin 5: Pulso en DEC-
 - ▶ Pin 6: Pulso en RA-
- Protección mediante opto-acopladores
- Conforme al estándar ST-4 (pero con análisis de guiado por software)

Otras características importantes

- No tiene buffer para imágenes
- No tiene control de temperatura
- No está refrigerada
- No incluye lente ni control de foco

Primeras capturas y análisis

- El objetivo era determinar que parámetros de configuración le enviaba el programa cliente (QGVideo5T para Windows) a la cámara para configurarla.
- Se sospechaba que las configuraciones se enviaban una vez y luego se recibían frames con dicha configuración.
- Se confirmó esto en los primeros análisis, pero algo pasaba cuando se conectaba la cámara por primera vez.
- Antes de iniciar la captura con el programa QGVideo5T y cuando la cámara se conectaba por primera vez, el led parpadeaba un par de veces y luego quedaba prendido. Luego la captura sucedía normalmente.
- Hubo algunos inconvenientes para capturar el tráfico antes de capturar.
- Una vez realizada la captura de este tráfico anómalo, resultó que no se correspondía ni con datos del sensor ni con configuraciones (al menos que pudieran determinarse a simple vista).

Primeras capturas y análisis (cont)

- Captura de un frame
- Captura de un WTF
- A través del análisis detallado del tráfico se pudo determinar

Ganancia (2 bytes por canal)

Fila y columna de comienzo

URB	0016	10:31:14.141	5.940211 s		Control Transfer
00000000	00 00 00 05 00 05 00 05 00 05 00 14 00 20 05 FF				
00000010	07 FF 00 8E 00 19 00 00 00 00 00 00 03 A9 00 03				
00000020	5F 00 00 19 FA 00 00 06 3C 01 0A 00 F3 C0 9E 00				
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00				

Horizontal y Vertical Blanking

Resolución(2048x1536)

Reconstrucción del firmware

- Los datos RAW del firmware

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum
- El firmware reconstruido en formato IHEX válido y listo para cargar con *FXLoad*

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum
- El firmware reconstruido en formato IHEX válido y listo para cargar con *FXLoad*
- Una vez reconstruido el firmware y cuando ya se estaba analizando el código desensamblado, el fabricante hizo disponible su versión del firmware además del código fuente de referencia del driver para Windows

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum
- El firmware reconstruido en formato IHEX válido y listo para cargar con *FXLoad*
- Una vez reconstruido el firmware y cuando ya se estaba analizando el código desensamblado, el fabricante hizo disponible su versión del firmware además del código fuente de referencia del driver para Windows : (

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum
- El firmware reconstruido en formato IHEX válido y listo para cargar con *FXLoad*
- Una vez reconstruido el firmware y cuando ya se estaba analizando el código desensamblado, el fabricante hizo disponible su versión del firmware además del código fuente de referencia del driver para Windows :(:/

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum
- El firmware reconstruido en formato IHEX válido y listo para cargar con *FXLoad*
- Una vez reconstruido el firmware y cuando ya se estaba analizando el código desensamblado, el fabricante hizo disponible su versión del firmware además del código fuente de referencia del driver para Windows :(:/ :)

Reconstrucción del firmware

- Los datos RAW del firmware
- El programa para cargarle las direcciones
- El script para el checksum
- El firmware reconstruido en formato IHEX válido y listo para cargar con *FXLoad*
- Una vez reconstruido el firmware y cuando ya se estaba analizando el código desensamblado, el fabricante hizo disponible su versión del firmware además del código fuente de referencia del driver para Windows :(:/ :) :D

El controlador - ¿Por qué en espacio de usuario?

¿Por qué?

- El controlador de referencia para la cámara QHY5 es en espacio de usuario
- Escribir en espacio de kernel es difícil
- El rendimiento alcanzado desde espacio de usuario no justifica un driver en espacio de kernel

El controlador - ¿Por qué en espacio de usuario?

¿Por qué?

- El controlador de referencia para la cámara QHY5 es en espacio de usuario
- Escribir en espacio de kernel es difícil
- El rendimiento alcanzado desde espacio de usuario no justifica un driver en espacio de kernel

Ventajas

- Escribir en espacio de usuario es más sencillo
- No hay que estar pendiente de cambios en la interfaz del kernel

El controlador - ¿Por qué en espacio de usuario?

¿Por qué?

- El controlador de referencia para la cámara QHY5 es en espacio de usuario
- Escribir en espacio de kernel es difícil
- El rendimiento alcanzado desde espacio de usuario no justifica un driver en espacio de kernel

Ventajas

- Escribir en espacio de usuario es más sencillo
- No hay que estar pendiente de cambios en la interfaz del kernel

Desventajas

- No hay dispositivo de video (`/dev/video0`)
- No pueden utilizarse programas existentes (hay que adaptarlos o escribir nuevos)

El controlador - Arquitectura de un vistazo

- Un tipo de datos que representa el dispositivo.
- Funciones exportadas para abrir, cerrar, configurar y obtener frames del dispositivo.
- Funciones privadas para comunicación con el dispositivo.

Las funciones exportadas

- `qhy5t_open()`: Crea las estructuras y abre la conexión con el dispositivo.
- `qhy5t_set_params()`: Guarda los parámetros de configuración.
- `qhy5t_program_camera()`: Envía los comandos de configuración guardados a la cámara.
- `qhy5t_start_capture()`: Inicia el thread de captura.
- `qhy5t_read_exposure()`: Lee un frame.
- `write_pgm()`: Escribe un frame a disco.
- `qhy5_timed_move()`: Inicia un comando de guiado.
- `qhy5_cancel_move()`: Cancela los movimientos de guiado.
- `qhy5t_stop_capture()`: Detiene el thread de captura.
- `qhy5t_close()`: Esta función cierra la conexión con el dispositivo y libera la memoria allocada para las estructuras.

Detalles de implementación

¿Preguntas al respecto?

Detalles de implementación

- La función `ctrl_msg()`
- La función `set_gain()`
- El proceso de adquisición.

El visor de un vistazo

```
declarar las estructuras
procesar los parámetros
inicializar las estructuras SDL
programar la cámara
iniciar la captura
while (True):
    procesar los eventos de teclado SDL
    realizar un requerimiento de un frame al controlador
    debayerizar el frame
    mostrar el frame en la surface SDL
    guardar el frame en un archivo (si esto es requerido)
```

¿Preguntas al respecto?

Detalles del visor

- Los comandos de teclado
- El debayerizado
- El crossair

Las pruebas de rendimiento

Resolución	Debayerizado (FPS)	Sin debayerizar (FPS)
320x240	71.96	90.73
640x480	18.97	36.05
800x600	12.38	25.98
1024x768	7.59	17.20
2048x1536	1.90	2.86

Tabla: Impacto del debayerizado en el frame rate.

Las pruebas de rendimiento(cont)

Resolución	Escritura Linux (FPS)	Escritura Windows 8 (FPS)
320x240	99.96	100.43
640x480	39.96	37.46
800x600	27.31	23.35
1024x768	19.18	16.35
2048x1536	5.88	3.3

Tabla: Comparación de tiempos de escrituras a disco para Windows y Linux.

Fotografías obtenidas

Veamos algunas fotos :)

Fotografías obtenidas

Veamos algunas fotos :)
... y después una demo :D