SOC_1202_T12_120221

Diseño software versión final

Versión: 20121130

Autor: Rafel Mormeneo Melich

Inicio	22/10/2012
Final	30/11/2012
Personas	Rafel Mormeneo Melich

Predictive
Control
Systems

Thinking Forward
www.tfxxi.com

VARIACIONES RESPECTO LA VERSIÓN ANTERIOR:

Versión anterior	Ninguna
Variación	1- Ninguna

1- Autor: Vacio. Descripción: Vacio.

1 Diseño software versión final

1. Objetivos

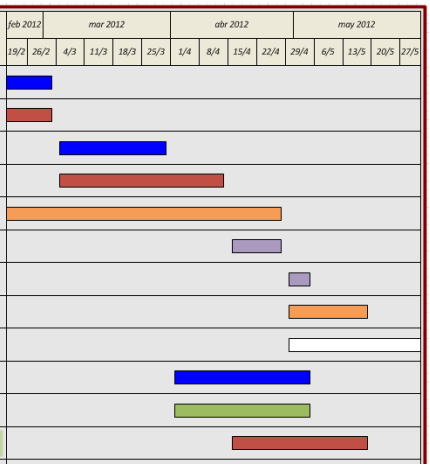
El objetivo de esta tarea consiste en desarrollar el software de la versión final del prototipo. Este software tiene que integrar todos los sistemas del SOC: comunicación CAN, comunicación RS-232 con el PC y con la cámara, acceso a la memoria SDRAM, descompresión de imágenes JPEG, y acceso a la memoria FLASH interna para grabar los parámetros de configuración.

Las funcionalidades que deberán implementarse son las siguientes:

- Comunicación con DAEN y ECON.
- Captura de imágenes.
- Transmisión de imágenes al ECON.
- Funciones de instalación.

2 Plan de desarrollo de la tarea

ID	Tarea	Descripción tarea	Responsable tarea	Inicio	Final	Duración
1	SOC_1202_T1_120221	Definir hardware cámara	Rafel	21/02/2012	02/03/2012	9d
2	SOC_1202_T2_120221	Definir el microcontrolador, familirizarse con el entorno de desarroyo, can, etc.	Fran	21/02/2012	02/03/2012	9d
3	SOC_1202_T3_120221	Diseñar prototipo	Rafel	05/03/2012	30/03/2012	20d
4	SOC_1202_T4_120221	Diseñar software prototipo	Fran/Ezio	05/03/2012	13/04/2012	30d
5	SOC_1202_T5_120221	Pruebas de algoritmos básicos	Víctor	21/02/2012	27/04/2012	49d
6	SOC_1202_T6_120221	Definición protocolo de comunicación Servidor/Cliente/SOC	Ezio/Victor/Fran	16/04/2012	27/04/2012	10d
7	SOC_1202_T7_120221	Implementación del protocolo de comunicación Servidor/Cliente/SOC	Ezio/Fran	30/04/2012	04/05/2012	5d
8	SOC_1202_T8_120221	Cambios en TFClient derivados de la aplicación de SOC	Victor/Fran	30/04/2012	18/05/2012	15d
9	SOC_1202_T9_120221	Diseño de algoritmos alternativos mejorados de visión.	Todos	30/04/2012	01/06/2012	25d
10	SOC_1202_T10_120221	Rediseño de hardware SOC	Rafel/Ezio	02/04/2012	04/05/2012	25d
11	SOC_1202_T11_120221	Diseño industrial SOC	Marc/Rafel/Oriol	02/04/2012	04/05/2012	25d
12	SOC_1202_T12_120221	Diseño del software de visión final	Fran/Ezio/Victor	16/04/2012	18/05/2012	25d



Dentro del proyecto SOC, esta tarea (Marcada en verde en el diagrama) ocupa un total de 25 días que van desde el 16/04/2012 hasta el 18/05/2012. Debido a la restructuración del proyecto SOC y al retraso de las tarea precedentes, esta tarea no ha empezado hasta el día 22/10/2012

3 Desarrollo de la tarea

Para el desarrollo de esta fase se han identificado las siguientes tareas:

- Desarrollo de la función de inicialización.
- Desarrollo de la funcionalidad para la captura de imágenes.
- Definición del protocolo de comunicación entre SOC – DAEN y entre SOC – ECON.
- Desarrollo de la funcionalidad de instalación.
- Test de todas las funcionalidades en el prototipo.

3.1 Desarrollo de la función de inicialización

Cómo su nombre indica, en esta función inicializamos todos los módulos necesarios para el funcionamiento del SOC. En primer lugar inicializamos el reloj a 60MHz, que es el máximo al que puede funcionar el microcontrolador. Habilitamos el clock para el módulo CAN.

Configuramos los pines de entrada salida estándar. LED bicolor, driver alimentación cámara y LEDs de flash.

A continuación iniciamos el UART para comunicación con el C429 a 115200bps. También inicializamos el UART para comunicación con el PC en caso que se haya definido el símbolo DEBUG.

Inicializamos la SDRAM y realizamos un test para comprobar que funciona correctamente. El test consiste en alocar un vector de 100 enteros dinámicamente, escribir valores del 0 al 99, leer estos valores y verificar el valor leído.

Inicializamos un timer que se encargará, en principio de hacer parpadear un LED. En un futuro podríamos encontrar otra funcionalidad para este timer.

Leemos los parámetros de configuración de la User Page de la FLASH interna. En caso no estén inicializados, grabamos los valores por defecto definidos en el fichero soc_conf.h.

Configuramos el bus CAN para que escuche el canal X a 50Kbps. La primera acción después de inicializar el dispositivo será reservar memoria para los Mensajes de Rx del CAN e iniciar el modo Rx.

Inicializamos las interrupciones de los UART y del timer. Para ello tenemos que definir los handlers de las interrupciones.

Iniciamos la librería JPEG.

En último lugar iniciamos el timer.

3.2 Desarrollo de la funcionalidad para la captura de imágenes

El módulo C429 se comunica con el microcontrolador a través del puerto serie (UART). Una imagen consiste en una serie de datos enviados a través del UART. El protocolo es el siguiente.

El master le envía a la cámara la secuencia de caracteres:

0x56 0x00 0x32 0x0C 0x00 0x0A 0xAA 0xAA 0xAA 0xAA 0xLL 0xLL 0xLL 0xLL 0x0B 0xB8

Dónde AA es la dirección (4 bytes) y LL es la longitud (4 bytes) del buffer que se quiere leer. La cámara responde con la secuencia:

0x76 0x00 0x32 0x00 0x00

Seguido de tantos bytes cómo se haya indicado en el campo LL del comando enviado. Al final vuelve a enviar la misma secuencia 0x76 0x00 0x32 0x00 0x00.

La recepción de la imagen se ha hecho mediante la interrupción del UART. Se ha definido una máquina de estados que espera la secuencia antes mencionada, luego lee el número de bytes correspondientes a la imagen y finalmente vuelve a leer la secuencia antes mencionada. Al final devuelve el control al proceso main para que haga la acción pertinente con la imagen (enviarla al PC, al ECON o calcular la holgura).

3.3 Definición del protocolo de comunicación entre SOC – DAEN y entre SOC – ECON

Hemos definido un nuevo protocolo de comunicaciones en el bus CAN para facilitar el desarrollo de varios dispositivos en el mismo bus. Este protocolo consiste en asignar SID distintos para cada par origen a destino de comunicaciones.

Para el SOC se han definido los siguientes SID:

Origen a Destino	SID
SOC a SPDR	0x083
SPDR a SOC	0x0C2
SOC a ECON	0x00B
ECON a SOC	0x0C1

El documento Especificaciones/Comunicación/SID.xlsx contiene un resumen de los SID del bus de campo. El directorio Especificaciones/Comunicación/SOC contiene los ficheros con los mensajes que se envían entre los distintos dispositivos y el SOC. El fichero Especificaciones/Protocolos/Protocolo_Transmision_Imagen ECON-SOC.docx contiene el protocolo de documentación.

El Atmel permite definir vairos Mob (Message Object) cada uno con un filtro diferente. De esta forma podemos filtrar distintos tipos de comunicación de forma sencilla.

Lo que no tiene el Atmel es una pila de mensajes recibidos de forma que mientras se está procesando un mensaje se debe deshabilitar la recepción de otros mensajes CAN. Esto produce muchas perdidas de paquetes ya que el procesado de los paquetes en algunos casos puede ser un proceso largo.

Para evitar este inconveniente se ha desarrollado una pila de mensajes software. Así, cuando recibimos un mensaje desactivamos las interrupciones, ponemos el mensaje en la pila, volvemos a activar las interrupciones y luego procesamos el mensaje. Con esto conseguimos tener el mínimo tiempo posible las interrupciones deshabilitadas y tenemos menos perdida de paquetes.

3.4 Desarrollo de la funcionalidad de instalación

En la instalación deberán configurarse varios parámetros:

1. DAEN asociado. Mediante el PC se asociará el SOC al DAEN correspondiente. Mientras el SOC no esté asociado a ningún DAEN irá elaborando una lista con los DAEN conectados al bus. El PC pedirá la lista de DAEN al SOC i el usuario decidirá a que DAEN asociarlo. Una vez asociado, el SOC envía un paquete al DAEN para informarle que tiene un SOC asociado y espera la confirmación de este. El SOC tendrá el mismo ID que el DAEN al que está asociado.
2. Ventanas de adquisición. Mediante el programa PC se le dirá al SOC dónde están las ROI dónde va a calcular las holguras. Estas posiciones se grabarán al SOC como parámetros de configuración en la User Page (FLASH interna).

Los parámetros de configuración se grabaran en la FLASH interna del Atmel, más concretamente en la User Page. Para debemos configurar correctamente el dispositivo y declarar una estructura de datos que se alojará en dicha sección de la memoria interna. Para ello utilizamos la siguiente instrucción,

```
__attribute__((__section__(".userpage"))) static nvram_parameters_t user_nvram_data;
```

Dónde se indica que la variable user_nvram_data se algergará en la sección .userpage definida en el linker script.

Ha surgido un problema con la escritura en la FLASH. En el principio del código la rutina de escritura memcpy funciona correctamente pero cuando se llama más adelante en el código, por ejemplo para escribir algun parámetro de configuración, esta deja de funcionar. Lo hemos solucionado modificando el fichero flashc.c. Concretamente se ha modificado la siguiente función

```
void flashc_issue_command(unsigned int command, int page_number)
{
	u_avr32_flashc_fcmd_t u_avr32_flashc_fcmd;

	//flashc_wait_until_ready();
	flashc_default_wait_until_ready();
	u_avr32_flashc_fcmd.fcmd = AVR32_FLASHC.fcmd;
	u_avr32_flashc_fcmd.FCMD.cmd = command;
	if (page_number >= 0) {
		u_avr32_flashc_fcmd.FCMD.pagen = page_number;
	}
	u_avr32_flashc_fcmd.FCMD.key = AVR32_FLASHC_FCMD_KEY_KEY;
	AVR32_FLASHC.fcmd = u_avr32_flashc_fcmd.fcmd;
	flashc_error_status = flashc_get_error_status();
	//flashc_wait_until_ready();
	flashc_default_wait_until_ready();
}
```

Se ha sustituido la llamada a la función flashc_wait_until_ready() por la flashc_default_wait_until_ready(). De hecho la primera no es más que un puntero a la segundo. Eso soluciona el problema y de esta forma funciona correctamente.

3.5 Test de todas las funcionalidades en el prototipo

Para realizar dicho test se toman varias imágenes con el SOC.

4 Conclusiones

Para realizar la versión final del código vamos a utilizar en Atmel Studio 6.0 en lugar del AVR32 Studio como veníamos haciendo hasta ahora. Esta decisión se debe a que Atmel no mejora el entrono de desarrollo basado en eclipse, sino que se decanta por el entorno basado en Microsoft Visual Studio.

Se ha desarrollado todo el software de captura y transmisión de imágenes. En total transferir una imagen del SOC al ECON cuesta, de media, 45 segundos y del ECON al server aproximadamente 20 segundos.

Esta tarea se ha completado en 107 horas (13 días y medio).