



Esteban Chiama <eechiama@gmail.com>

Info II TPO - Consulta PERFORMANCE

2 messages

Esteban Chiama <eechiama@gmail.com>

3 December 2018 at 14:59

To: Augusto Santini <santini_augusto@hotmail.com>

Hola Augusto,
Perdoname que hoy te llené de mails. Este es una pregunta de código, y sobre filosofía.

Resulta que nuestro ADC para 5 canales no está funcionando. La estrategia que usamos fue la siguiente:

- Timer Int. -> Adc start
- (Adc Int.[flag=1] -> Adc_start) repetidos hasta hacer todos los canales

Donde **flag** le indica a las MDE que hay un dato disponible del adc.

Desde las MDE:

```
if(flag){  
    // categoriza la muestra en vectores, uno para cada canal, secuencialmente. cuando hay 10 por  
    canal, filtro de mediana  
}
```

Contexto: al hacer la calibración para obtener los rangos de movimiento de cada dedo.

Problema: canal conectado al dedo índice escribe sobre el vector del meñique, y otros ejemplos así.

Hay canales que no aparecen en ningún lado también.

Si recordás, tengo una app en MATLAB para ver cada canal. Esto confirma lo que sucede. Si muevo el pote del índice lo veo en los datos que corresponden al meñique.

Y así con muchos dedos, es un quilombo.

Sospecho que el problema es nuestro approach. Hay mucho código para categorizar correctamente cada muestra y en el medio el ADC debe tener listas varias. Entonces se "pierden" muestras y se escriben en cualquier lado.

(las soluciones evidentes, que hayamos escrito mal un índice en las asignaciones o algo, ya las revisamos)

Solución propuesta

Para que la MDE sea más rápida: Eliminar la Int. del Timer. Disparar ADC Int. por flanco de timer match. El prox. paso sería DMA para el ADC pero no me animo.

Además: Re-hacer el driver de ADC. La lógica que ahora implemento es la siguiente:

Las muestras se guardan en un **vector de muestras** donde según el índice se corresponde con un canal.

El **flag** que avisa ahora se pone en alto cuando se llena el vector de muestras.

Decido seguir utilizando los data registers de cada canal, en lugar de cambiar al global. Para accederlos rápido sin hacer muchos ifs o un switch (como hacía antes), ahora tengo un **vector de data registers** y el índice es el canal seleccionado.

Ahora, para las **primitivas** tengo que organizar estas muestras hasta que tenga 10 por canal y ahí hacer el filtro.

Antes, tenía 5 vectores de muestras.

Ahora, quiero hacerlo bien. Investigando me metí un quilombo: "Array of structures" o "structure of arrays", a lo que le sumo la opción de array 2D.

Array of Structures:

```
typedef struct muestras {
    ch0;
    ch1;
    etc.. (5 canales)
} muestras_t;
muestras_t Data_Para_Filtrar [SIZE];
```

** size es el tamaño para hacer la mediana. 10 muestras

Structure of Arrays:

```
typedef struct muestras {
    uint16_t ch0 [SIZE];
    uint16_t ch1 [SIZE];
    etc..
} muestras_t;
```

Array Multidimensional:

Pero si quisiera hacer un **driver genérico**, donde cambiar la cantidad de entradas sea solamente cambiar un define, debería utilizar algo así:

```
uint16_t muestras[CANALES][SIZE];
```

** CANALES: cantidad de canales del adc a utilizar

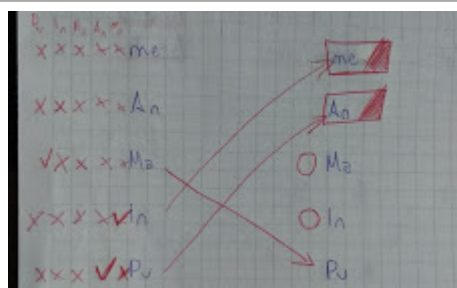
Según la internet, en C los array multidimensionales se implementan con memoria consecutiva en los elementos de la fila. Osea que acceder a todas las muestras de un canal específico sería rápido.

La idea es que una vez que se llenen las muestras aplicar el filtro, y ahí tengo una muestra efectiva de todos los canales.

Perdoname que te mande este problema, debés estar ocupado. Solamente quería saber tu opinión sobre si ALGO de lo que estoy diciendo es razonable o estoy encarando para cualquier lado. Capaz tenés alguna sugerencia sobre que tipo de "data handling" usar, cuál sería más óptimo o cosas así.

Desde ya muchísimas gracias por tu tiempo, si estás muy ocupado no te preocupes.
Esteban

pd: adjunto una imagen que muestra lo que está sucediendo con los dedos.



dedos.jpeg
85K

Augusto Santini <santini_augusto@hotmail.com>
To: Esteban Chiama <eechiama@gmail.com>

3 December 2018 at 17:43

Hola Esteban no hay problema, siempre intento hacerme un hueco que en el peor de los casos es a la noche, para responder consultas.

Lo más simple es aplicar un array multidimensional, la última opción que me mandaste. Te recomiendo no tocar cosas que andan (por ejemplo, el tema de las interrupciones de Timer, además no toma nada de tiempo eso, no ganás demasiada performance). La explicación de por qué les está andando mal, tiene que ser que están mezclando los índices. Usen #defines para que esto no les pase... No importa qué numero es qué canal ni qué dedo, simplemente hagan algo como:

```
#define CANAL_0 2
#define CANAL_1 5
#define etc...
```

```
#define CANAL_INDICE CANAL_0
#define CANAL_MAYOR CANAL_1
#define etc...
```

Entonces despues directamente indexan el array multidimensional como:

```
// Asumiendo que tienen alguna variable donde guardan qué canal está activo en este momento
```

```
// Y una variable donde guardan que número de muestra es...
```

```
// (El número de muestra es relativo a los 5 dedos a la vez. Se incrementa cada vez que muestrean los 5 canales)
```

...

```
uint16_t filtro_mediana(uint8_t seleccion_canal){
    switch(canal){
        case CANAL_INDICE:{
            // Recibe un array de uint16_t y cuántas muestras tiene para realizar la mediana
            return filtro_mediana(array_multidimensional[CANAL_INDICE], N_MUESTRAS);
            break;
        }
        ...etc...
    }
}
```

Espero haberme explicado bien, cualquier cosa respondeme pero mandame el código también así puedo ver en donde puede llegar a estar el problema.

Saludos.

From: Esteban Chiama <eechiama@gmail.com>

Sent: Monday, December 3, 2018 2:59 PM
To: Augusto Santini
Subject: Info II TPO - Consulta PERFORMANCE

[Quoted text hidden]