

PROYECTO FINAL, LA PC COMO CONTROLADORA DE PROCESOS.
SOFTWARE PARA MANEJO DE SISTEMA DE CONTROL
DE TAREAS DE RESONANCIA MAGNETICA NUCLEAR
2011

INTEGRANTES:

GOMEZ ELFI ADRIAN
VOLARIK MARIANO
MONTI RAUL

PROFESORES:

ZANINETTI WALTER
MARQUEZ CARLOS

INTRODUCCION:

rmnController.exe es un programa escrito en C/C++ capaz de controlar un sistema compuesto por un generador de pulsos (pp2), un conversor A/D (modelo AD9852), un sintetizador digital de frecuencias (dds). El fin de este equipo es generar señales con frecuencia y fase programables para un equipo de resonancia magnética nuclear, así como también adquirir los datos posteriores al experimento.

El programador de pulsos dirigirá el funcionamiento del resto de las partes componentes del sistema. El dds generará las señales y el conversor A/D tomará los resultados guardando hasta 64K adquisiciones por canal en sus dos canales de 12bits. La comunicación se logra vía el puerto paralelo, usando la interfaz de hardware disponible para comunicarse con cada una de las 3 placas por separado. El software se usará sobre una pc con sistema operativo Windows XP Sp3.

COMPORTAMIENTO GENERAL Y FORMA DE USO:

El programa funciona a través de una interfaz gráfica la cual asegura que los datos sean ingresados correctamente. Por lo tanto para usarlo es necesario simplemente seguir los pasos que esta indica. Sin embargo es necesario tener en cuenta los siguientes puntos:

- 1- Se pide como dato de entrada el programa de pulsos. Este deberá ser un archivo .txt redactado de acuerdo al apartado _.
- 2- La repetición del experimento mas de una vez (usando el campo “Número de repeticiones”) provoca que se pise el archivo que guarda los resultados de cada adquisición. Por lo que solo quedará al guardada la adquisición correspondiente a la ejecución final.

Internamente una ejecución normal del programa sigue los siguientes pasos:

- 1- Se lee y revisa la hoja de pulsos comprobando su correctitud sintáctica. Se usa para ello la función parse() perteneciente al módulo “parser”. Esta función devuelve una estructura 'instruction_sheet' con toda la información obtenida de la hoja de pulsos.
- 2- Se agrega la información de tiempos de cada pulso y delay a la instruction_sheet previamente creada. Esta información viene dada a partir de un archivo redactado según el apartado _ y debe corresponder a la hoja de pulsos parseada en el punto 1. Para todo esto se usa la función instruction_sheet_config_instructions() del módulo 'instruction_sheet'.
- 3- Se configura y habilita el dds. Para ello es necesario saber cuales son las frecuencias de trabajo deseadas para el dds. Si bien el dds tiene la posibilidad de cargar 2 frecuencias distintas, el programa solo admite la utilización de la frecuencia número 1. Todo esto se hace utilizando las funciones dds_config(), dds_set_freq() y dds_enable() todas del módulo 'dds'.
- 4- Carga las fases en la RAM de fases para su posterior utilización desde el dds. La información de las fases se toma a partir de la estructura instruction_sheet. Se usa para esto la función dds_load_phases_ram() del módulo 'dds'.

5- Se carga el programa de pulsos en el pp2 usando la estructura instruction_sheet a partir de la función pp2_load_program().

6- Se configura el A/D usando la función ad_config() del módulo 'ad'. Para ello se requieren los siguientes datos: cantidad de muestras por segundo; tamaño en K de la cantidad de adquisiciones; el modo en que se desea adquirir (solo esta disponible en este programa el modo continuo).

7- Se dispara el programador de pulsos. Para ello se usan las funciones pp2_microprocessor_mode_enabled() y pp2_launch_pulse_sequence() ambas pertenecientes al módulo 'pp2'.

8- Se espera el resultado de la adquisición usando la función adquiere() perteneciente al módulo 'ad' y opcionalmente se los guarda en el archivo de salida especificado usando la función ad_to_file(). Este último paso es obligatorio si se desea poder graficar mas adelante los resultados ya que el programa esta diseñado para comunicarse con la interfaz y el graficador a través de los archivos que estos generan.

9- Se llama al programa "Graficador.exe" el cual interpreta el archivo de salida del A/D y lo grafica. (Notar que esto permite colocar en el directorio cualquier programa que sepa interpretar los datos guardados en el archivo de salida del A/D siempre y cuando se le dé el nombre "Graficador.exe").

10- Termina el programa.