

Práctica 10

Generador de Frecuencia y Convertidor Analógico-Digital del ATmega1280

Objetivo: Mediante esta práctica el alumno aprenderá la programación y uso avanzado del Temporizador 0 y 2 del microcontrolador ATmega1280, así como el uso del Convertidor Analógico-Digital.

Material:

- Computadora Personal (con AVR Studio)
- Tarjeta T-Juino.
- Componentes Electrónicos.

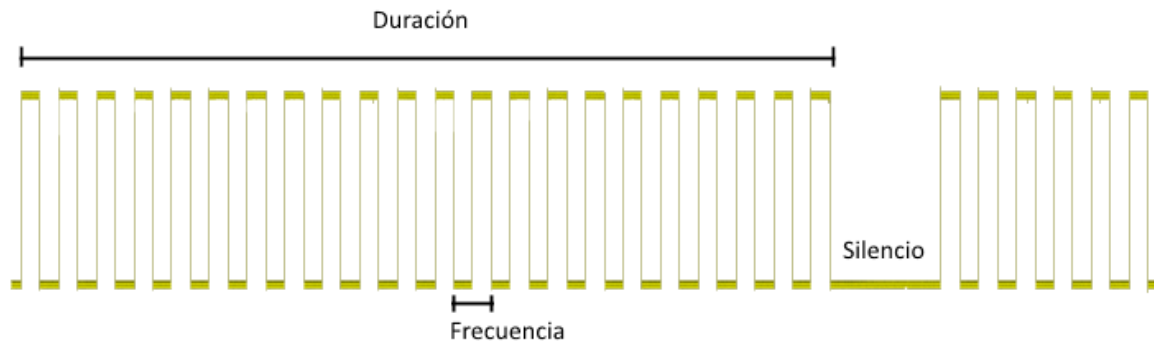
Equipo: - Computadora Personal con USB, AVRStudio y WinAVR

Teoría:

- Programación del Timer2 del microcontrolador como generador de frecuencia
- Programación y uso del ADC
(Diagrama, Funcionamiento, Registros de configuración y operación)

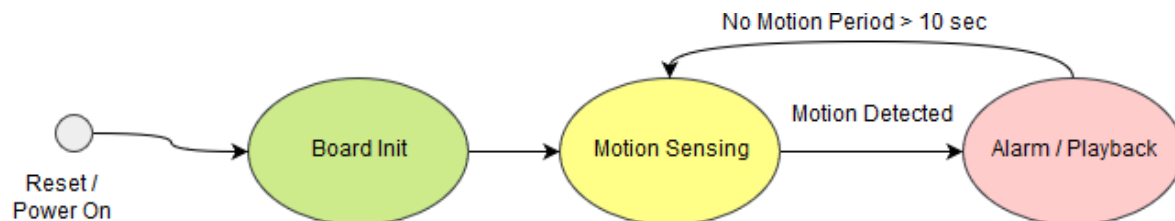
Desarrollo:

El propósito de la práctica es la implementación de una “alarma” en base a generación de tonos mediante el generador de frecuencia; y “detección de movimiento” mediante el uso del ADC. En la siguiente figura se puede visualizar lo que llamaremos un *tono*:



Donde cada tono que podemos generar tendrá una frecuencia arbitraria en un rango posible de ~60Hz hasta ~15KHz, con solo 254 frecuencias intermedias. Y con volumen arbitrario de cada tono, donde el volumen máximo (100%) corresponderá al 50% del PWM.

Y con este mecanismo implementado se deberá implementar el siguiente comportamiento:



Para lograr lo anterior se requiere reutilizar el código del Timer0 de la práctica anterior, y anexar la siguiente función:

- 1) void **Timer2_Gen_Tone**(uint16_t tone, uint8_t volume)
Si "tone" es mayor que 0 entonces, inicializa y habilita el Generador de Frecuencia del Timer2 para generar un tono aproximado y volumen al que se recibe como parámetro. De lo contrario se requiere deshabilitar el Generador, generando de esta forma el silencio (0 lógico en el pin OC2B).

Los tonos de la "alarma" pueden ser los que ustedes quieran, como ejemplo se podría utilizar el siguiente patrón: 1KHz por 500ms, silencio por 500ms, 1KHz por 500ms, silencio por 500ms, y así sucesivamente. Como recomendación, usen volumen de 25% o menos, para que no sea molesto a los que tengan alrededor.

Y posteriormente se requieren implementar las siguientes funciones de configuración y operación del ADC:

- 2) void **ADC_Ini** ()
Esta función inicializa para 8 bits de resolución y habilita el ADC del microcontrolador de forma genérica. Encontrar el desplazamiento (offset) de la medición y almacenarla.
- 3) uint8_t **ADC_Read**(uint8_t channel)
Esta función lo que realiza es una lectura del ADC usando el canal correcto y retornando el valor de 8 bits acorde a la aplicación (ver Figura 2), compensando el desplazamiento de la medición.
- 4) uint8_t **motion_Detected**(void)
Esta función utiliza ADC_Read() y en base a las últimas muestras determina si ha detectado algún movimiento.

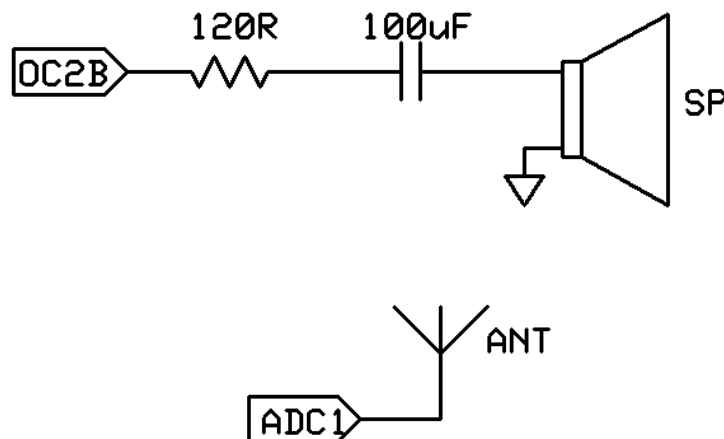


Figura 2. Esquemático

Comentarios y Conclusiones.

Bibliografía.