UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS QUIMICAS E INGENIERIA

Microcontroladores y microprocesadores

Practica 11. Generador de Frecuencia mediante los Temporizadores del uC ATmega1280

Alumno: Caudillo Sanchez Diego

Matricula: 1249199

Docente: Jesus Adan Garcia Lopez

Fecha de entrega: 15/mayo/2020

Teoria

Teoría básica de música

Primero que nada, se tiene que tener a la mano el concepto de música. Esta es una forma racional y lógica de organizar el sonido y el silencio en composiciones musicales. La música es un arte, y como tal, sigue criterios estéticos para expresar sentimientos y emociones. A continuación se listan los elementos del sonido:

- **Altura:** hace referencia a cuán grave o agudo es un sonido. La mayor parte de los instrumentos musicales producen sonidos con altura definida, aunque hay ciertos instrumentos de percusión que no poseen esta propiedad.
- **Duración:** se refiere al tiempo que se prolonga un sonido. La combinación de duraciones entre diferentes sonidos y silencios da origen al ritmo.
- **Intensidad:** está relacionada con el volumen. Hace referencia a que tan fuerte o suave es un sonido.
- **Timbre:** es la propiedad que hace que un sonido de una guitarra suene como una guitarra o el sonido de un violín suene como un violín. En general se dice que los sonidos son ondas, pero no son ondas puras, sino que se componen de muchas ondas llamadas armónicos, los armónicos de un sonido definen su tiembre. Aunque hay otros aspectos involucrados en un timbre, por ejemplo el ataque o la resonancia.

Elementos de la música.

- **Melodía:** se refiere a la organización sucesiva de los sonidos uno después del otro. Se entiende la melodía como la organización de las alturas de los sonidos en un sentido horizontal.
- **Armonía:** se refiere a los sonidos producidos simultáneamente. La armonía se produce cuando varias persona cantan al tiempo diferentes notas o tocan diversas notas en sus instrumentos. También se produce cuando un pianista presiona simultáneamente varias teclas en un piano. La armonía se relaciona con la organización de las alturas de los sonidos en un sentido vertical la melodía en un sentido horizontal.
- **Ritmo:** se refiere a la distribución de los sonidos en el tiempo. Cuando escuchamos una pieza musical percibimos una sensación de movimiento. Esa sensación es el resultado de la forma en que las distintas notas se ha distribuido en la pieza.
- **Métrica:** se refiere a patrones regulares en el ritmo que se hace evidente a través de los acentos naturales.
- **Dinámica:** hace referencia a la combinación de intensidades de los sonidos.

Formas musicales

Las formas musicales son estructuras generales para orientar una composición. Fundamentalmente, las formas musicales definen la organización de los de motivos y temas en una composición. Algunas formas musicales pueden definir también aspectos como el estilo, la métrica, las textura, la agógica o la estética general de una composición.

Forma binaria

La forma binaria es una de las formas de estructurar las composición musicales. Como su nombre lo indica, las composiciones en forma binaria se estructuran en dos partes (A y B). Estas composiciones fueron muy comunes en el periodo barroco.

Forma ternaria

La forma ternaria es similar a la forma binaria. Su estructura es A-B-A. es decir, que luego de la parte B se repite la parte A. La forma ternaria queda de la siguiente manera:

- **Sección A:** se nos presenta una serie de motivos. La sección se encuentra en la tonalidad principal de la obra.
- **Sección B:** se nos presentan motivos que contrastan con los de la sección A aunque con algunas similitudes. La sección comienza en otra tonalidad, puede ser una tonalidad cercana, homónima o relativa. Luego retorna a la tonalidad principal.
- **Seccion A:** también llamada reexposición, se repite la sección A.

Rondo

El Rondo es una forma musical en la que se repite constantemente un estribillo. La estructura del rondo sería A-B-A-C-A-D-A, se puede prolongar indefinidamente.

Forma sonata

Esta forma es una estructura generalmente utilizada en el primer movimiento de las contras. Se usa también en algunas sinfonías y conciertos. La forma sonata es mucho más compleja tiene secciones que se dividen en diferentes subsecciones. Las tres principales secciones se denominan exposición desarrollo y reexposición. A continuación, un resumen de las tres parte de la forma sonata:

- **Exposición:** en la exposición se presentan los motivos y temas que estarán presentes durante toda la obra. La exposición se divide, a su vez, en dos partes, A y B.
    - **Parte A:** se encuentra en la tonalidad principal, aunque puede modular hacia una tonalidad relativa u homónima.
    - **Parte B:** se encuentra en la tonalidad dominante.

- **Desarrollo:** se conforma de melodías elaboradas a partir de los motivos presentado en la exposición. A menudo se trata de variación. Al final del desarrollo se hace un rallentando para retomar a la exposición. En el desarrollo suele haber múltiples modulaciones hacia diferentes tonalidades.
- **Reexposición:** aquí se retorna a la exposición con una pequeña diferencia: la Parte B se encuentra en la tonalidad principal, no se modula a la dominante.

Programación del Timer2 del microcontrolador con generación de frecuencia

La bandera de Overflow (TOV2) se activa cada que el contador alcanza el tope. Y si las interrupciones están activadas, el manejador de rutinas de interrupción pueden ser usadas para actualizar el valor de comparación.

En el modo PWM rápido, la unidad de comparación permite la generación de formas de onda PWM en el pin OC2x. Configurar COM2x1: 0 bits a dos producirá un PWM no invertido y una salida PWM invertida se puede generar configurando COM2x1: 0 en tres. TOP se define como 0xFF cuando WGM2: 0 = 3, y OCR2A cuando WGM2: 0 = 7 El valor real de OC2x solo ser visible en el pin del puerto si la dirección de datos para el pin del puerto se configura como salida. La forma de onda PWM se genera configurando (o borrando) el registro OC2x en la comparación entre OCR2x y TCNT2, y borrando (o configurando) el registro OC2x en el ciclo del reloj del temporizador El contador se borra (cambia de TOP a BOTTOM).

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC2A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match, set OC2A at BOTTOM, (non-inverting mode).
1	1	Set OC2A on Compare Match, clear OC2A at BOTTOM, (inverting mode).

La frecuencia PWM para la salida se puede calcular mediante la siguiente ecuación:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

La variable N representa el factor de preescalador (1, 8, 32, 64, 128, 256 o 1024). Los valores extremos para el registro OCR2A representan casos especiales al generar un PWM salida de forma de onda en el modo rápido PWM. Si el OCR2A se establece igual a BOTTOM, la salida ser un pico estrecho para cada ciclo de reloj de temporizador MAX + 1. Establecer el OCR2A igual a MAX resultará en una salida constantemente alta o baja (dependiendo de la polaridad de

la salida establecida por el COM2A1: 0 bits). Se puede lograr una salida de forma de onda de frecuencia (con un ciclo de trabajo del 50%) en modo PWM rápido configurando OC2x para alternar su nivel lógico en cada coincidencia de comparación (COM2x1: 0 = 1). La forma de onda generado tendrá una frecuencia máxima de $f_{oc2} = f_{clk\_I\_O}/2$ cuando OCR2A se establece en cero. Esta característica es similar a la palanca OC2A en modo CTC, excepto la función de doble búfer de la Salida La unidad de comparación está habilitada en el modo rápido PWM.

Lectura y escritura de de Datos en la Memoria de Programa

Hay tres grupos de memoria en los microcontroladores utilizados en las placas Arduino, en el caso particular que utilizamos el 2560.

- Memoria flash (espacio del programa), es donde se almacena el boceto Arduino.
- SRAM (memoria estática de acceso aleatorio) es donde el boceto crea y manipula variables cuando se ejecuta.
- EEPROM es un espacio de memoria que los programadores pueden usar para almacenar información a largo plazo.

La memoria flash y la memoria EEPROM no son volátiles (la información persiste después de desconectar la alimentación). SRAM es volátil y se perderá cuando se apague y vuelva a encender.

Hay que tomar en cuenta que la memoria flash (PROGMEM) solo se puede completar en el momento de grabación del programa. No puede cambiar los valores en la memoria flash después de que el programa haya comenzado a ejecutarse.

A continuación se muestra una tabla con la cantidad de memoria que cuentan los microcontroladores Arduino:

	ATMega168	ATMega328P	ATmega1280	ATmega2560
Flash (1 kByte used for bootloader)	16 kBytes	32 kBytes	128 kBytes	256 kBytes
SRAM	1024 bytes	2048 bytes	8 kBytes	8 kBytes
EEPROM	512 bytes	1024 bytes	4 kBytes	4 kBytes

Una cosa que se puede notar en el cuadro anterior, es que hay mucha más memoria Flash (programa) que SRAM disponible. Cuando creamos variables como lo siguiente:

char message [] = "Hello World";

Está copiando 11 bytes (1 char = 1 byte, más nulo de terminación) de la memoria del programa en SRAM antes de usarlo. 11 bytes no es mucha memoria en un grupo de 1024 bytes, pero si el boceto requiere algunas estructuras de datos grandes e inmutables, como una gran cantidad de texto para enviar a una pantalla o un *lookup table*, por ejemplo, usar la memoria flash (memoria de programa) directamente para almacenamiento, puede ser la única opción. Para lograr hacer esto, se utiliza la palabra clave PROGMEM.

PROGMEM almacene datos en la memoria flash (programa) en lugar de SRAM. La palabra clave PROGMEM es un modificador variable, debe usarse solo con los tipos de datos definidos en *pgmspace.h*. Le dice al compilador que "ponga esta información en la memoria flash", en lugar de en la SRAM, donde normalmente almacenaría.

Si bien PROGMEM podría usarse en una sola variable, en realidad solo vale la pena si tiene un bloque de datos más grande que necesita ser almacenado, que generalmente es más fácil en una matriz, (o otra estructura de datos).

Usar PROGMEM también es un procedimiento de dos pasos. Después de obtener los datos en la memoria Flash, se requieren métodos especiales (funciones), también definidos en la biblioteca pgmspace.h, para volver a leer los datos de la memoria del programa en SRAM, para que podamos hacer algo útil con ellos.

Para ello, esta librería tiene declarada funciones y macros para la lectura de los datos almacenados en la memoria de programa.

En la siguiente imagen, se muestran algunas macros para lectura de datos almacenados en FLASH.

```
#define pgm_read_byte_near(address_short) __LPM((uint16_t)(address_short))
#define pgm_read_word_near(address_short) __LPM_word((uint16_t)(address_short))
#define pgm_read_dword_near(address_short) __LPM_dword((uint16_t)(address_short))
#define pgm_read_float_near(address_short) __LPM_float((uint16_t)(address_short))
#define pgm_read_ptr_near(address_short) (void*)__LPM_word((uint16_t)(address_short))
#define pgm_read_byte_far(address_long) __ELPM((uint32_t)(address_long))
#define pgm_read_word_far(address_long) __ELPM_word((uint32_t)(address_long))
#define pgm_read_dword_far(address_long) __ELPM_dword((uint32_t)(address_long))
#define pgm_read_float_far(address_long) __ELPM_float((uint32_t)(address_long))
#define pgm_read_ptr_far(address_long) (void*)__ELPM_word((uint32_t)(address_long))
```

Comentarios y conclusiones

Con el desarrollo de esta práctica se aprendió a utilizar el Timer2 con el generador de frecuencia, utilizando el pwm. La práctica fue bastante interactiva ya que con ejemplo de la vida real, como es la música, se pueden generar melodías con tan solo sonido y silencio. Obviamente esto tiene un acercamiento meramente matemático, pues debido a que las notas musicales tienen una cierta frecuencia que las distingue una de otras es posible crear piezas actualizando la frecuencia del microcontrolador con la ayuda de los temporizadores, para determinar el final de una nota y volver a tocar otra, haciendo una secuencia que como resultado genera música.

Referencias

https://es.wikibooks.org/wiki/Teor%C3%ADa_musical/Conceptos_b%C3%A1sicos/Texto_completo

https://www.componiendomivida.com/teoria-musical-principiantes-ritmo/

https://angelsguitar.com/conceptos-basicos-de-teoria-musical-parte-i/

https://www.nongnu.org/avr-libc/user-manual/group__avr__pgmspace.html

https://www.arduino.cc/reference/en/language/variables/utilities/progmem/

https://playground.arduino.cc/Learning/Memory/