Práctica 11

Uso del Convertidor Analógico Digital del ATmega1280

Objetivo: Mediante esta práctica el alumno aprenderá la programación y uso básico del convertidor analógico digital del microcontrolador ATmega1280.

Material: 1 – Tarjeta T-Juino
1 – Cable USB

Equipo: Computadora Personal con USB, AVRStudio y WinAVR

Teoría: – Programación y uso del ADC (Diagrama, Funcionamiento, regs. de conf. y operación).
– Funcionalidad y caratcteristica de la fotoresistencia.

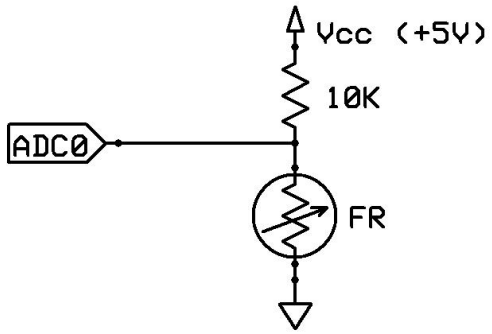

Figura 1. Diagrama para fotoresistencia

Desarrollo: Modifique el programa de la Práctica 10 para que el volumen de las notas que se generan sea controlada por la intensidad de luz. Para esto se deberá diseñar e implementar las siguientes funciones de configuración y operación considerando el diagrama de la Figura 1. Además el ancho de pulso (PWM) de la terminal de OC2B sera acorde a la luminosidad captada por el fotoresistencia.

1) void **ADC_Ini** ( )
Esta función inicializa para 8 bits de resolución y habilita el ADC del microcontrolador de forma generica. Encontrar el desplazamiento (offset) de la medición y almacenarla.

2) uint8_t **ADC_Read**(uint8_t channel)
Está 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 1), compensando el desplazamiento de la medición.

3) void **ADC_MinMax**(uint8_t channel)
Función que captura el rango de valores, encuestando primero por el mínimo.

4) #define **ADC_Normalize**(value) ??
Macro que retorna el valor normalizado de 0 a 100, tomando en cuenta la relacion de los valores maximos y minimos.

5) void **Timer2_Set_Volume**(uint8_t volume){
Ajusta el ancho de pulso que es producido sobre la terminal OC2B. El rango del valor de entrada sera de 0 a 100.

Reutilizando el código elaborado en la práctica anterior y seguir el flujo de **Listado 1**.

Listado 1:

```
#define PHOTORESISTOR PORTA0

int main(void)
{
    UART0_Ini();
    UART0_AutoBaudRate();

    Timer0_Ini();
    ADC_Ini();
    ADC_MinMax(PHOTORESISTOR);

    while(1){
        if (UART0_available()){
            if(UART_getchar() == 'p'){
                Timer2_Play(ImperialMarch,sizeof(ImperialMarch)/sizeof(struct note));
            }
        }
    Timer2_Set_Volume(ADC_Normalize(ADC_Read(PHOTORESISTOR)));
    }
    return 0;
}
```

Comentarios y Conclusiones.

Bibliografía.