#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <TimerOne.h>

#define MICRO 53 //pin sensor de sonido

#define ACT\_S 51 //activar foco con sonidop

#define LDR 52//pin sensor luz

#define ACT\_L 50 //pin salida sensor luz

LiquidCrystal\_I2C lcd (0x27, 20, 4);

unsigned sec = 0;

void Sen\_Sound();

void show (const char\* str, int CoorX = 0, int CoorY = 0) {

    if (CoorX >= 0 && CoorY >= 0)

        lcd.setCursor(CoorX, CoorY);

    lcd.print(str);

}

void setup () {

    lcd.init();

    lcd.setCursor(0, 0);

    lcd.backlight();

    Timer1.initialize(1000000);

    Timer1.attachInterrupt(Sen\_Sound);

    pinMode(LDR, INPUT);

    pinMode(ACT\_L, OUTPUT);

    pinMode(MICRO, INPUT);

    pinMode(ACT\_S, OUTPUT);

    digitalWrite(ACT\_S, HIGH);

    digitalWrite(ACT\_L, HIGH);

    show("Act Sonido: OFF");

    show("Act Luz: OFF", 0, 1);

}

void loop () {

    if (digitalRead(MICRO)) {

        digitalWrite(ACT\_S, LOW);

        show("Act Sonido: ON ");

    }

    else

      show("Act Sonido: OFF");

    if (digitalRead(LDR) == 0) {

        digitalWrite(ACT\_L, LOW);

        show("Act Luz: ON ", 0, 1);

    }

    else {

        digitalWrite(ACT\_L, HIGH);

        show("Act Luz: OFF", 0, 1);

    }

}

void Sen\_Sound()

{

        if (digitalRead(ACT\_S) == LOW){

        sec++;

        if (!(sec % 10)) {

            digitalWrite(ACT\_S, HIGH);

            //

            sec = 0;

          }

        }

}

Se incluyen las bibliotecas y se definen los pines que se va utilizar para entrada y salida.

Se necesitara un variable volatile para analizarla en la interrupción, en el setup se inicializa todo lo que se va requerir en el programa, es decir los pines se ponen como entrada y salida y se inicia con los focos apagados, se hicieron 2 funciones 1 para mostrar el estado de los focos, otra para la interrupcion.

Ahora cada 10 segundos según la interrupción declarada en setup se va a apagar el foco, si es que este encendido en caso de que este apagado sin importar que tantas veces se active la interrupción no se hará nada, ahora si se detecta un ruido el sensor mandara un estado de alto por lo que el foco se encenderá, pasando 10 seg se apagara, se hizo con un timer para que el foco del sensor de luz se pueda encender y apagar de manera independiente del sensor de sonido

Al superar cierto grado de luz el foco se enciende y se cubre de la luz se apaga el foco, esto es porque el módulo de luz Arduino solo puede enviar salidas digitales, es decir 0 y 1.