

UNIVERSIDAD DE ORIENTE.

NUCLEÓ ANZOÁTEGUI.

ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS

DEPARTAMENTO DE INGENIERIA.



TUTOR

PROFESOR:

ING. ALFONSO ALFONSI

EVALUACIÓN 1: ASIGNACIÓN ESQUEMA DE ENTRADA Y SALIDA, PWM, ENTRADA DE POTENCIOMETRO, INTERRUPTORES, TEMPORIZADORES

CORREA LUIS	CARDOZA MARIA
CI N° 19840230	CI N° xxxxxxxx
ASIGNATURA: ARQUITECTURA AVANZADA DEL COMPUTADOR	SECCIÓN: 01

CONTENIDO

1.	INTRODUCCIÓN.....	3
2.	DESARROLLO (EXPERIENCIAS, DIAGRAMAS DE LOS CIRCUITOS Y CÓDIGOS).....	3
3.	DISCUSIÓN DE RESULTADOS.....	9
4.	CONCLUSIONES.....	9
5.	REFERENCIAS	9
6.	NORMAS	9

1. INTRODUCCIÓN

En el presente trabajo se realizan la carga de los programas en el hardware denominado Arduino Uno¹ y se verifican los circuitos:

- a) El esquema de entrada y salida digital
- b) Salida de PWM.
- c) Entrada de potenciómetro
- d) Interrupciones
- e) Temporizador

El presente trabajo muestra los esquemas y algoritmos utilizados para realizar la asignación.

2. DESARROLLO (EXPERIENCIAS, DIAGRAMAS DE LOS CIRCUITOS Y CÓDIGOS)

a) Entrada y salida

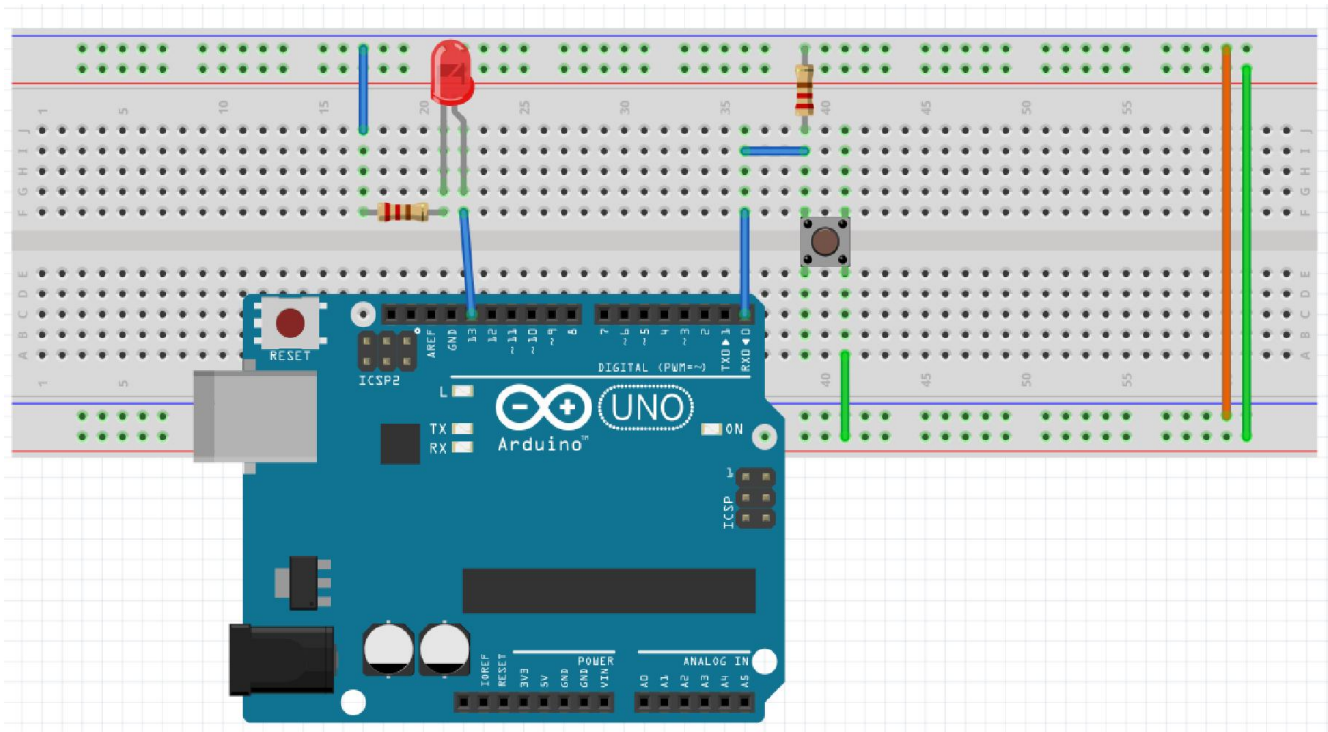


Figura N° 1. Entrada y Salida de Arduino Uno.

```
int entrada = 0;
int salida = 13;

void setup(){
  pinMode(entrada, INPUT);
```

¹ Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia)

```

pinMode(salida, OUTPUT);
}
void loop(){
    int valor = digitalRead(entrada);
    if(valor != 0){
        digitalWrite(salida, HIGH);
    }else{
        digitalWrite(salida, LOW);
    }
}

```

b) Salida PWM

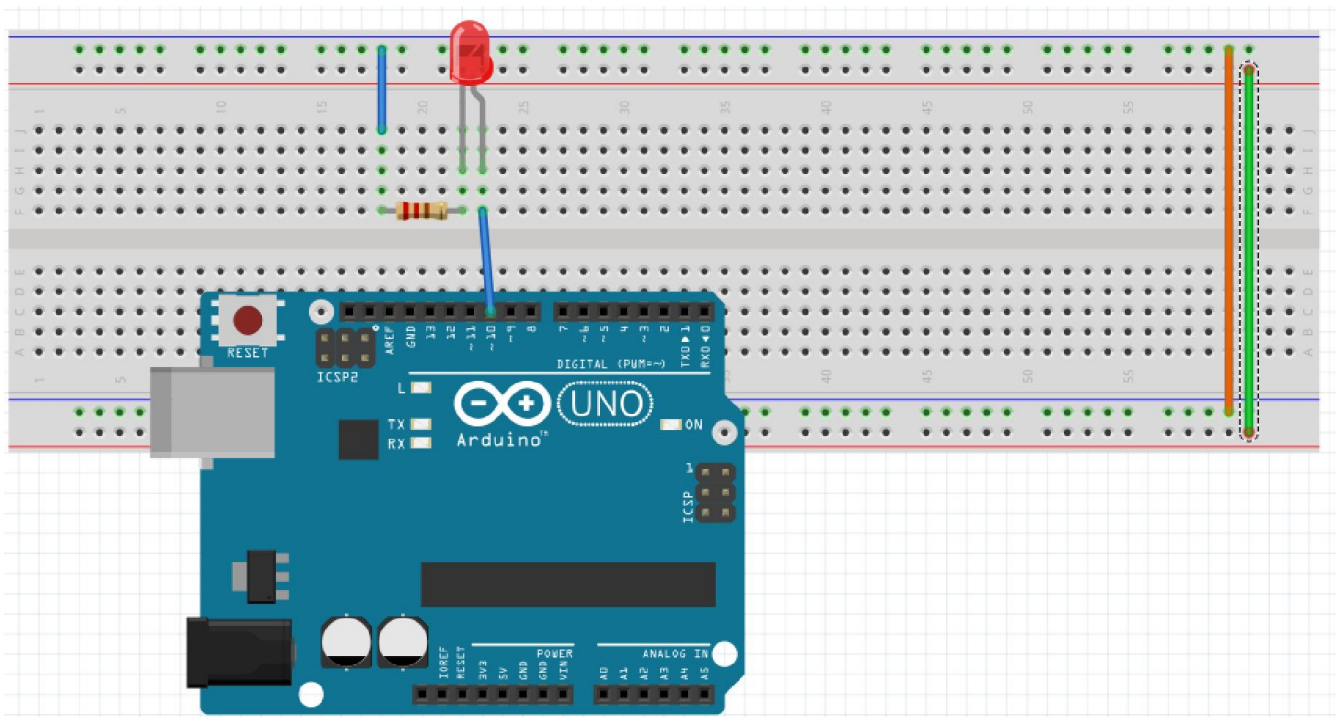


Figura N° 2. Salida PWM de Arduino Uno.

```

int digPin = 10; // pin digital 10
void setup() {
    pinMode(digPin, OUTPUT); // pin en modo salida
}

```

```

void loop(){
    digitalWrite(digPin,HIGH); // asigna el valor HIGH al pin
    delay(500); // espera medio segundo
    digitalWrite(digPin,LOW); // asigna el valor LOW al pin
    delay(500); // espera medio segundo
}

```

c) Entrada de potenciómetro:

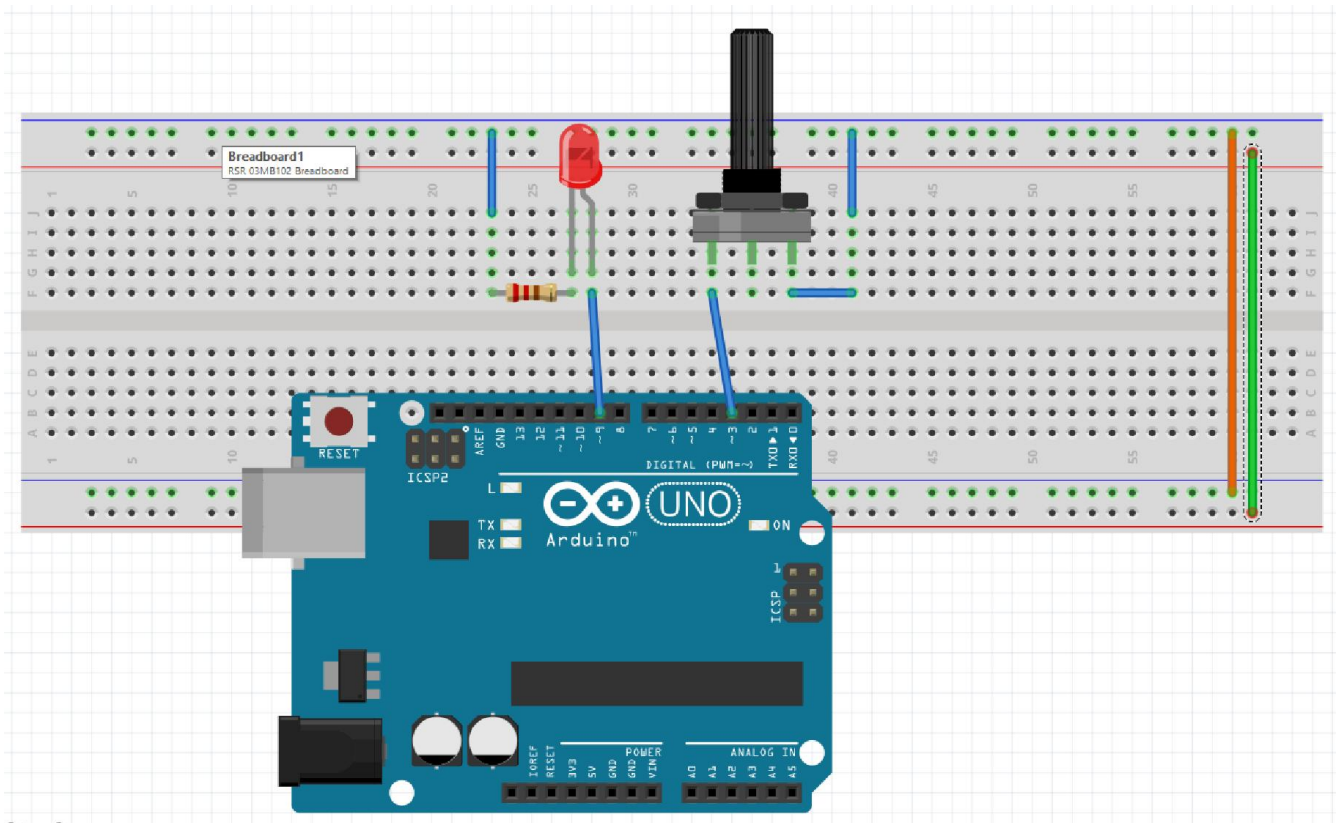


Figura N° 3 Entrada Potenciómetro Salida PWM.

```

int ledPin = 9; // LED connected to digital pin 9
int analogPin = 3; // potentiometer connected to analog pin 3
int val = 0; // variable to store the read value

void setup(){
    pinMode(ledPin,OUTPUT); // sets the pin as output
    pinMode(analogPin,INPUT);
}

```

```

}

void loop(){
    val = analogRead(analogPin); // read the input pin
    analogWrite(ledPin, val / 4); // analogRead values go from 0 to 1023, analogWrite values from 0 to 255
}

```

d) Interrupciones:

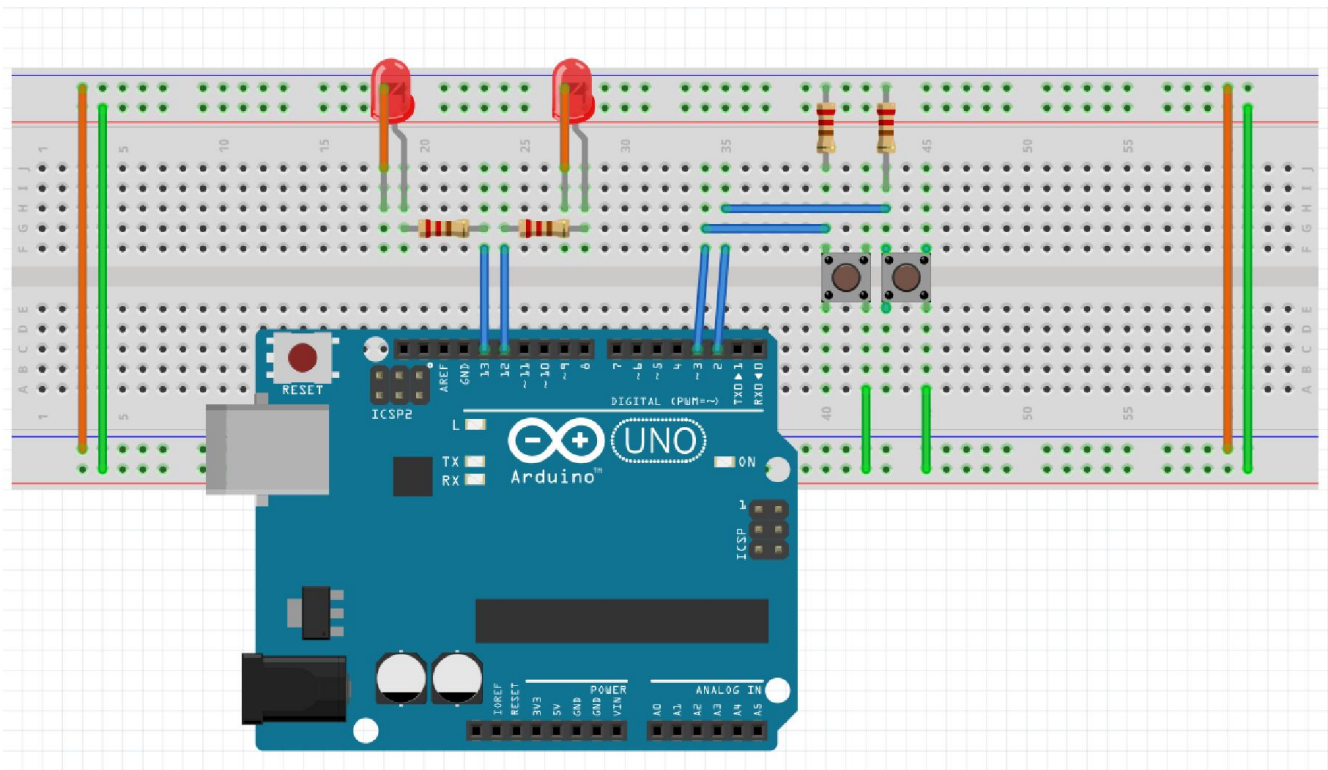


Figura N° 4. Interrupciones Arduino Uno.

```

int pin = 13;
int pin2 = 12;
volatile int estado = LOW;
volatile int estado2 = LOW;
//el atmega328p solo tiene 2
// el 0 pin digital 2, el 1 en el pin digital 3
void setup(){
    pinMode(pin, OUTPUT);
    pinMode(pin2, OUTPUT);
    attachInterrupt(0, parpadeo, CHANGE); //CHANGE CAMBIO
}

```

```

attachInterrupt(1,parpadeo2,CHANGE);
}
void loop(){
  digitalWrite(pin,estado);
  digitalWrite(pin2,estado2);
}
void parpadeo(){
  estado = !estado;
}
void parpadeo2(){
  estado2 = !estado2;
}
}
e) Temporizador.

```

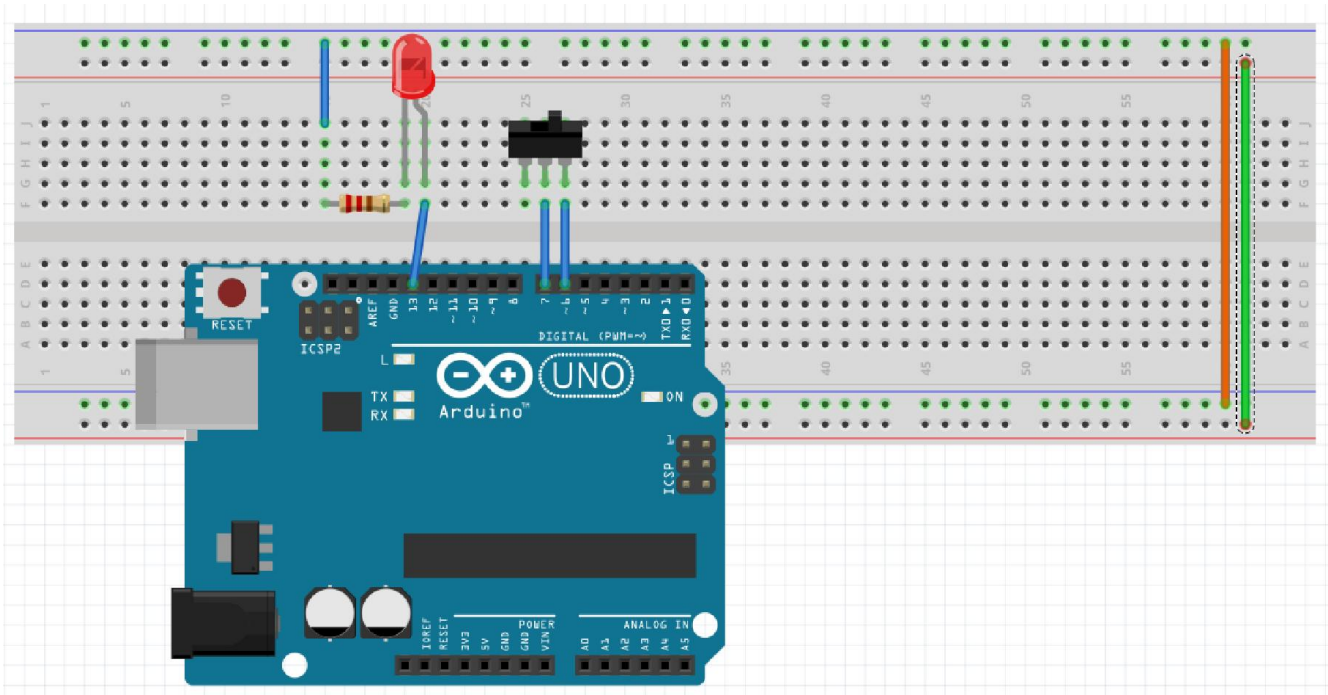


Figura N° 5. Temporizadores en Arduino Uno.

```

#define Pecho 6
#define Ptrig 7
long duracion,distancia;

void setup() {

```



```

Serial.begin(9600);    // inicializa el puerto serial a 9600 baudios

pinMode(Pecho,INPUT);  // define el pin 6 como entrada (echo)
pinMode(Ptrig,OUTPUT); // define el pin 7 como salida (trigger)
pinMode(13,1);         // Define el pin 13 como salida
}

void loop() {
  digitalWrite(Ptrig,LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig,HIGH); // genera el pulso de trigger por 10ms
  delayMicroseconds(10);
  digitalWrite(Ptrig,LOW);

  duracion = pulseIn(Pecho,HIGH);
  distancia = (duracion/2) / 29;    // calcula la distancia en centimetros

  if (distancia >= 500 || distancia <= 0){ // si la distancia es mayor a 500cm o menor a 0cm
    Serial.println(" --- ");          // no mide nada
  }else {
    Serial.print(distancia);          // envia el valor de la distancia por el puerto serial
    Serial.println("cm");             // le coloca a la distancia los centimetros "cm"
    digitalWrite(13,0);               // en bajo el pin 13
  }

  if(distancia <= 10 && distancia >= 1){
    digitalWrite(13,1);               // en alto el pin 13 si la distancia es menor a 10cm
    Serial.println("Alarma.....");   // envia la palabra Alarma por el puerto serial
  }

  delay(400);                         // espera 400ms para que se logre ver la distancia en la consola
}

```


3. DISCUSIÓN DE RESULTADOS

- 3.1 En los resultados se pudo comprobar el uso del Arduino uno al utilizarlo con switches, potenciómetros, sus salidas PWM y sus dos interrupciones.

COMPONENTES UTILIZADOS:

Arduino Uno

Potenciómetro

Resistencias

Leds

Switches

Cables

Software: Arduino Ide, Fritzing.

ACCIÓN Y RESULTADO:

Se enviaron señales a través del Arduino y el mismo envió señales de respuesta

- 3.2 Utilice el método delay en el Arduino ide para el control del sensor ultrasónico.

COMPONENTES UTILIZADOS:

Arduino Uno

Resistencias

Leds

Cables

Sensor Ultrasonico

Software: Arduino Ide, Fritzing.

ACCIÓN Y RESULTADO:

Se utilizó el sensor ultrasónico HC-SR04 para determinar la existencia de un objeto cerca del Arduino Uno.

4. CONCLUSIONES

- Se comprueba el funcionamiento del Arduino uno al enviar señales digitales en respuesta a otras entradas digitales.
- Se comprueba el funcionamiento del Arduino uno al enviar señales pwm para prender y apagar un led.
- Se comprueba el funcionamiento del Arduino uno al enviar señales pwm en respuesta a una señal recibida de un potenciómetro
- Se comprueba el funcionamiento del Arduino uno que cuenta con dos interrupciones externas.
- Se comprueba el funcionamiento del Arduino uno al poder controlar el tiempo de envío de señales con el método delay.

5. REFERENCIAS

- Óscar Torrente Artero., (2013). Arduino Curso Práctico de formación.

6. NORMAS

- Normas Apa: Para redacción de documentos