

**Laboratorio de**  
**Fundamentos de Instrumentación Biomédica, Gpo 2.**  
**Profesor: Armando Salomón Hernández Delgado M.I.**  
**Lab6: Amplificador de Biopotenciales – Diseño de ganancia programable.**

**Objetivo:** Crear un Amplificador para biopotenciales con ganancia programable empleando técnicas digitales (Arduino y resistencia con control digital).

**Trabajo Previo:**

1. Conocer el funcionamiento de la resistencia digital acoplado al amplificador de instrumentación implementado en la práctica 5. El rango de ganancias requerido es de 100 a 500.

**Material:**

1 resistencias  $1k\Omega$  y  $10k\Omega$ , 1 resistencia de  $100\Omega$ . 1 Arduino, cables BNC, Amplificador de instrumentación Alambrado o en su defecto, un amplificador monolítico (AD620), 1 potenciómetro digital [X9C103P](#) (10 Kohms).

**Introducción:**

El potenciómetro digital contiene 100 niveles o 'pasos' en los que se puede colocar el tap central, comenzando desde una resistencia mínima de 40 Ohms (nivel 0) hasta 10kOhms (nivel 99), con un incremento de 101 Ohms por paso.

El amplificador de Instrumentación de la figura 1 puede tener una ganancia programable si controlamos el valor de la resistencia por ejemplo con una resistencia digital y un microcontrolador.

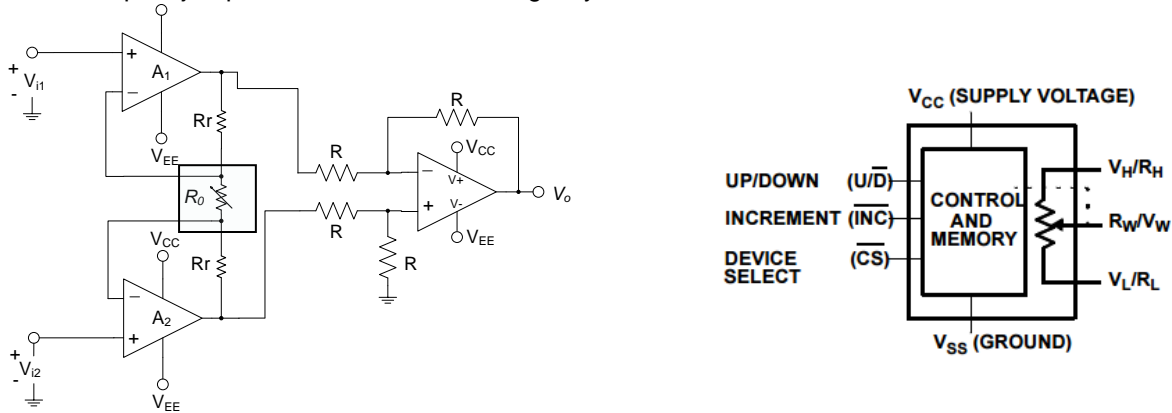


Figura 1. Amplificador de Instrumentación.

a) Si se conecta el potenciómetro digital sustituyendo la resistencia  $R_0$  La ganancia del amplificador de instrumentación  $G = \frac{2R_r}{R_G} + 1$ , se puede representar en función del número de niveles del potenciómetro y el valor de las resistencias  $R_r = 27k\Omega$ . Por lo tanto, la Tabla I representa algunos valores de ganancia para los niveles de resistencia digital, considerando.

Nivel	$R_G [\Omega]$	Ganancia
0	40	
1	141	
2	242	
3	343	
4	444	

Tabla I. Ganancia en función del nivel del wiper.

Podemos representar un modelo de la resistencia de ganancia como una ecuación lineal  $R_G = 101(\text{Nivel}) + 40$ . En consecuencia la ganancia máxima con  $R_{Gmin} = \underline{\hspace{1cm}}$  es  $G_{max} = \underline{\hspace{1cm}}$  y la ganancia mínima con  $R_{Gmax} = \underline{\hspace{1cm}}$  es  $G_{min} = \underline{\hspace{1cm}}$ .

Expresando la ganancia en función del nivel del wiper tenemos :

ec. 1:

$$G = \frac{2R_r}{101(Nivel) + 40} + 1$$

Si se desea una ganancia  $G$  específica entre  $G_{max}$  y  $G_{min}$  se determina el valor de  $R_G$  como:

ec.2

$$R_G = \frac{2R_r}{G - 1}$$

Y ésta en función del nivel del wiper del potenciómetro como:

ec 3.

$$101(Nivel) + 40 = \frac{2R_r}{G - 1}$$

Para una Ganancia determinada, se obtiene el nivel requerido como:

ec. 4.

$$Nivel = \frac{1}{101} \left[ \frac{2R_r}{G - 1} - 40 \right]$$

El valor de nivel calculado debe ser un valor entero.

b) Si  $G_{min} = 100$  y  $G_{max} = 500$ , teóricamente  $R_{G100} = 545\Omega$  ,  $R_{G500} = 108\Omega$  . Empleando el mismo potenciómetro digital, se hacen las siguientes modificaciones de diseño, para obtener las resistencias anteriores:

Mantener una resistencia fija de  $R_{G500} = 110\Omega$  para  $G_{max} = 500$ . Considerando que la resistencia equivalente de dos resistencias en paralelo es aproximadamente la resistencia más pequeña, se puede hacer este arreglo de una resistencia  $\sim 500\Omega$  en paralelo con la resistencia del potenciómetro que va desde  $40\Omega$  hasta  $10k\Omega$ , dando una resistencia en total de  $37\Omega$  hasta  $476\Omega$ . Colocada en serie con la resistencia  $R_{G500} = 110\Omega$  se obtiene una resistencia máxima de  $576\Omega$  generando una  $G_{min} = 94$ .

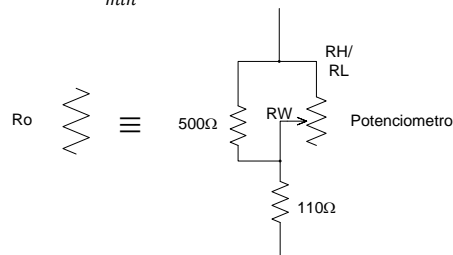


Figura 2. A.I. con ganancia variable de 100 a 500 con potenciómetro digital.

### Desarrollo:

1. Conecte el potenciómetro digital como se indica en la figura 3, mida el voltaje en el wiper con el osciloscopio. Empleando el programa base implemente el control del potenciómetro con la tarjeta Arduino. Compruebe el funcionamiento. Enviando un comando "0,x" se coloca el wiper en el nivel 0, por lo que en el wiper se medirá 0V. Un comando "1,x", incrementa el valor del wiper y un comando "2,x" lo decremента. La 'x' puede ser 0 o un valor indistinto. Experimente con los comandos adicionales: "3,n" para incrementar n pasos la resistencia, "4,n" para decrementar n pasos la resistencia, "5,n" para fijar la resistencia en el nivel n y "6,x" para fijar la resistencia máxima.

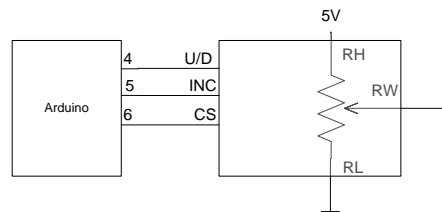


Figura 3. Conexión del potenciómetro digital con Arduino.

2. Tomando como ejemplo el análisis en la parte (a) obtenga una expresión para representar el número de pasos del potenciómetro para una ganancia determinada entre 100 y 500. Esta expresión se empleará en el siguiente ejercicio.

3. Implemente el A.I. con ganancia variable empleando el circuito de la figura 2. Genere una señal **senoidal de aproximadamente 100 mV@100Hz** aplicada a un divisor de voltaje con resistencias  $10k\Omega$  y  $1k\Omega$ . El voltaje de entrada al A.I. se toma de la resistencia de  $1k\Omega$  y se ajusta hasta tener 10 mV aprox.

Implemente un programa en Arduino que haga lo siguiente: recibirá un conjunto de comandos de una PC a través del puerto serie, y dependiendo del comando recibido se determina la acción con una sentencia de control switch-case. Tome como ejemplo el programa proporcionado.

Si hay dato disponible en puerto serie

Leer dato que llega por puerto serie.

Si el comando es válido se ejecuta la acción correspondiente

Comandos válidos (ver lista de comandos)

Si es comando invalido, no se hace nada.

Lista de comandos

Comando	Argumento	Acción
0	X	Cambia a Resistencia minima
1	X	Incrementa resistencia un paso
2	X	Decrementa resistencia un paso
3	N	Incrementa resistencia N pasos
4	N	Decrementa resistencia N pasos
5	N	Establece resistencia en el paso N
6	N	Cambia a Resistencia máxima
7	G	Fija Ganancia ~ (100-500)*

\*Empleando la expresión del punto 2, considerar solo valores enteros de número de nivel.

Compruebe el resultado de la ganancia establecida midiendo la amplitud de la señal de salida en el osciloscopio.

## Resultados.

Explique los resultados obtenidos en cada implementación.

## Conclusiones.

## Referencias.

### Código de ejemplo:

```
// conexion de Terminales del Arduino al potenciómetro digital
#define UD 4
#define INC 5
#define CS 6
#define clk 1 // duracion del pulso

void setup(){
  Serial.begin(115200);
  pinMode(UD, OUTPUT);
  pinMode(INC, OUTPUT);
  pinMode(CS, OUTPUT);
}

void potUp(int n){ // realiza n incrementos
  digitalWrite(CS, LOW); // habilita pot
  digitalWrite(UD, HIGH); // Incrementar w iper

  for(int i=0; i<n; i++){
    digitalWrite(INC, HIGH); // flanco de bajada
```

```

    delay(clk);          // retardo auxiliar
    digitalWrite(INC,LOW);
    delay(clk);
}
digitalWrite(CS,HIGH); // deshabilita pot
}

void potDow n(int n){    // realiza n incrementos
    digitalWrite(CS,LOW); // habilita pot
    digitalWrite(UD,LOW); // Decrementar w iper

    for(int i=n; i>=1; i--){
        digitalWrite(INC,HIGH); // flanco de bajada
        delay(clk);          // retardo auxiliar
        digitalWrite(INC,LOW);
        delay(clk);
    }
    digitalWrite(CS,HIGH); // deshabilita pot
}

void loop (){
    int i, cmd, arg;
    int val,oldp=0;

    if (Serial.available()>0){
        cmd = Serial.parseInt(); // lee valores enteros y separa cuando encuentra coma
        arg = Serial.parseInt();
        if (Serial.read() == '\n') { // si encuentra nueva linea continua
            Serial.println("cmd, arg");
            Serial.print(cmd,DEC); Serial.print(','); Serial.println(arg,DEC);

            switch(cmd){
                case 0 :
                    potDow n(100);
                    Serial.println("Resistencia min");
                    oldp = 0;
                    Serial.print("level "); Serial.println(oldp,DEC);
                    break;
                case 1: // incrementa un paso
                    if (oldp < 100){
                        oldp += 1;
                        potUp(1);
                        Serial.println("up");
                    }
                    Serial.print("level "); Serial.println(oldp,DEC);
                    break;
                case 2 : // decrementa un paso
                    if (oldp > 0){
                        oldp -= 1;
                        potDow n(1);
                        Serial.println("down");
                    }
                    Serial.print("level "); Serial.println(oldp,DEC);
                    break;
                case 3 : // incrementa n pasos
                    if (oldp < 100){
                        potUp(arg);
                        oldp += arg;
                        Serial.print(arg); Serial.println(" steps up");
                    }
                    Serial.print("level "); Serial.println(oldp,DEC);
                    break;
                case 4 : // decrementa n pasos
                    if (oldp > 0){
                        potDow n(arg);
                        oldp -= arg;
                        Serial.print(arg); Serial.println(" steps down");
                    }
                    Serial.print("level "); Serial.println(oldp,DEC);
                    break;
                case 5 : // un nivel determinado
                    val = abs(arg-oldp);
                    if(arg>oldp){
                        if ((oldp+val) < 100){

```

```

        potUp(val);
        oldp += val;
    }
}
else {
    if ((oldp-val)>0){
        potDown(val);
        oldp -= val;
    }
}
Serial.print("nivel "); Serial.println(oldp,DEC);
break;
case 6 : // resistencia maxima
    potUp(100);
    Serial.println("Resistencia max");
    break;
case 7 : // Escribir rutina para fijar ganancia

    break;
default: Serial.println("Comando/argumento no reconocido");
    break;
} //switch
} // if (Serial.read() == '\n')
} //if (Serial.available()>0)
} // void loop()

```