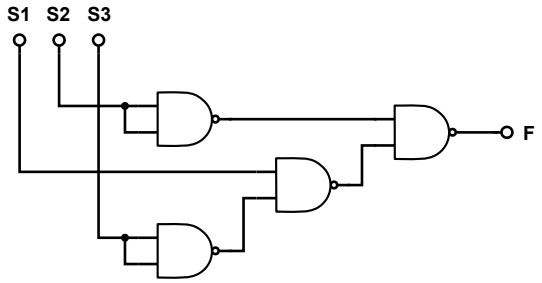
124 – Sistemas Electrónicos Santiago de Compostela 19/06/2022

Solucións Primeira Proba Parte A (Galego)

OPCIÓN A

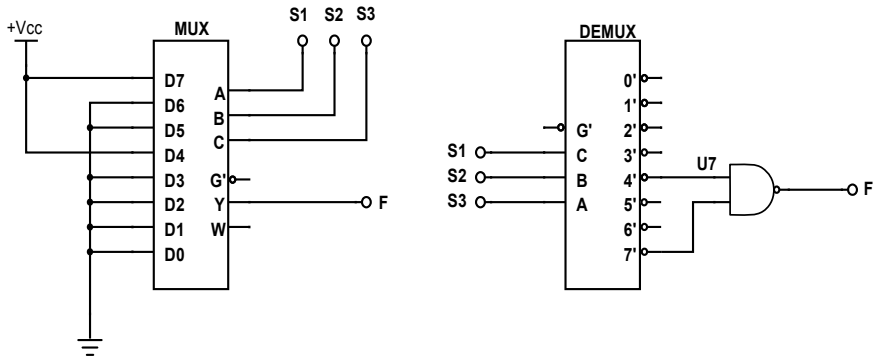
Exercicio 1 da Opción A:

Apartado a:



Apartado b:

Unhas posibles solucións serían:



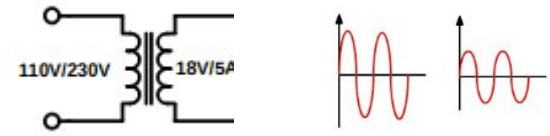
Exercicio 2 da Opción A:

Apartado a:

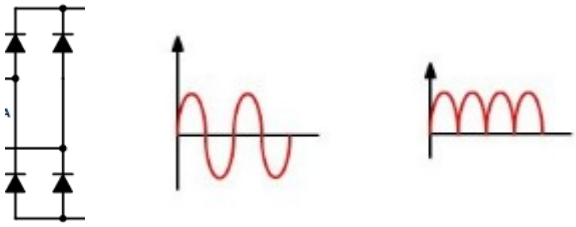
Convirte o sinal da rede eléctrica da entrada (alterno senoidal) nun sinal de continua constante (e estable fronte a variacións na carga) de valor variable. // Fonte de alimentación lineal regulable/variable/axustable.

Apartado b:

Bloque de transformación:

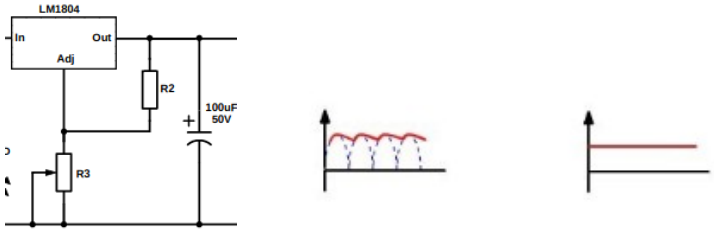


Bloque de rectificación:

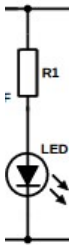


Bloque de filtrado:

Bloque de regulación/estabilización con valor de saída axustable:



Bloque adicional de indicación de alimentación (ON) sen efecto na forma do sinal:



Apartado c:

En modo DC, xa o que o modo AC elimina o nivel de continua.
En modo AC veríamos o sinal sen nivel de continua, polo que o veríamos centrado en 0V, podendo observar o grao de rizado presente no sinal de saída cunha escala V/div axeitada.

Apartado d:

Valor máximo visualizable: 4V

V_{REF}=1.25V (folla de características); desprezamos $I_{ADJ.}$

$$I_{R2} = I_{R3} = \frac{V_{REF}}{R2} = \frac{1,25V}{R2}$$

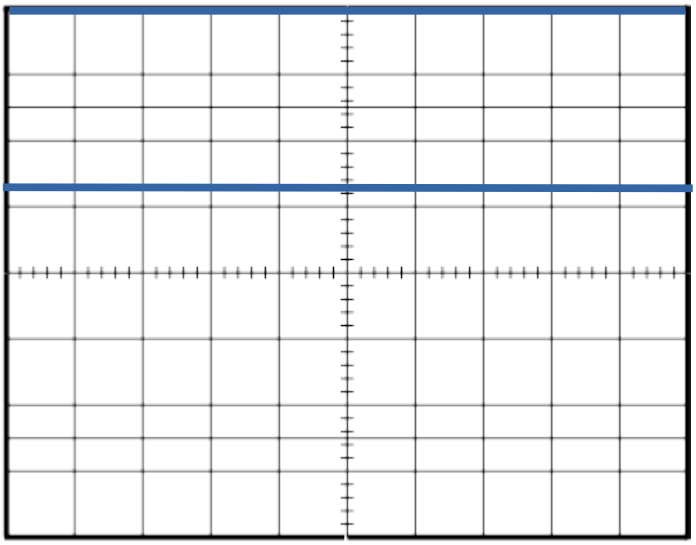
$$V_{R3} = I_{R3} \cdot R_3 = 1,25 \cdot \frac{R3}{R2}$$

$$V_{Out} = V_{REF} + V_{R3} = 1,25 \cdot (1 + \frac{R3}{R2})$$

$$R3_{max} \Rightarrow V_{Out} = 4V = 1,25 \cdot (1 + \frac{R3_{max}}{R2}) \Rightarrow \frac{R3_{max}}{R2} = \frac{4}{1,25} - 1 = 2,2$$

Posible elección: R2=1kΩ, R3=2,2kΩ (valor máximo do potenciómetro)

Valores a representar: 1,25V (R3= 0kΩ), 4V (R3= 2,2kΩ)



Canle A= 1V/div; Canle B= 1V/div; Time = 20 ms/div

Apartado e:

Tomamos caída nos díodos rectificadores: 0,7V

Caída no díodo LED: 2,4V para 20mA (folla de características). Tomamos 20mA como I_{MAX}

$$R_{LED,min} = \frac{24,06V - 2,4V}{20mA} = 1,08k\Omega \rightarrow 1,2k\Omega \pm 5\%$$

(I_{LED} > 10mA para V_{IN} > 14,4V => apagado imperceptible para o ollo humano)

Apartado f:

I_{max} debe garantizar o funcionamento do regulador

$$V_{rpp,max} = (V_{2,max} - 2 \cdot V_D) - (V_{REF} + V_{DROP}) = 18 \cdot \sqrt{2} - 2 \cdot 0,7 - (1,25 + 1,5) = 21,31V$$

Datos extraídos da imaxe, da folla de características e suposición V_D= 0,7V

Do rizado máximo, a máxima corrente extraída do condensador:

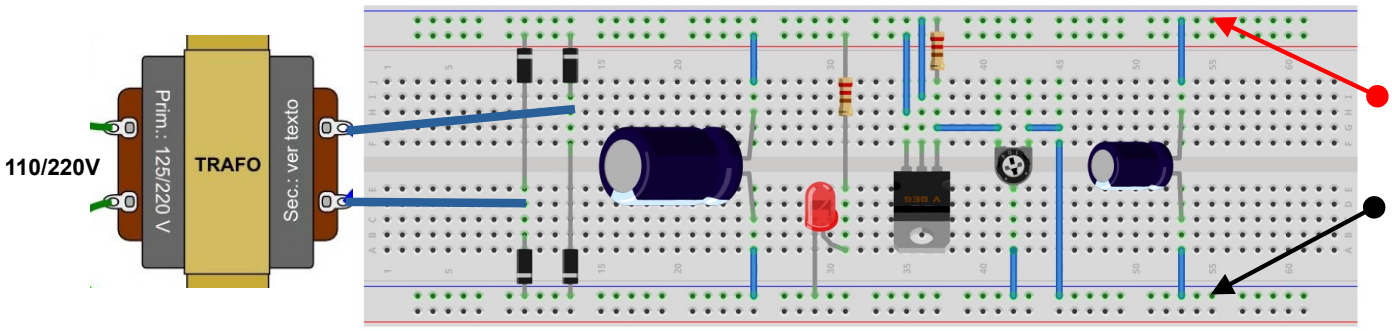
$$V_{rpp} = \frac{I_{CC}}{f_{riz} \cdot C} \Rightarrow I_{CC} = V_{rpp,max} \cdot f_{riz} \cdot C = 21,31 \cdot 100 \cdot 2,2mF = 4,69A$$

$$I_{LOAD,max} = I_{CC} - I_{LED} = 4,69 - 0,02 = 4,67A$$

I_{ADJ} (120μA max) e I_{R2} (1,25mA) desprezables

Apartado g:

Unha posible solución sería:



Exercicio 3 da Opción A:

Apartado a:

PC1 → PC2
PC1 → R1→ (212.1.2.2) R3 → (212.1.2.130) R4 → PC2
PC2 → PC1
PC2 → R4 → (212.1.2.129) R3 → (212.1.2.67) R5 → (212.1.2.65) R2 → (212.1.1.129) R1 → PC1
Dado que existe un camiño para ir e outro para voltar, PC1 e PC2 poden intercambiar tráfico sen necesidade de facer ningún cambio nas táboas de encamiñamento

Apartado b:

PC1 → PC3
PC1 → R1 → (212.1.1.130) R2 → ? (no hai ruta)
É necesario engadir en R2 unha ruta, además das que xa ten
R2 → (212.1.2.67) R5 → ?? (non hai ruta)
É necesario engadir en R5 unha ruta

Router	Destino	Máscara	Gateway
R2	212.1.3.0	255.255.255.0	212.1.2.67
R5	212.1.3.0	255.255.255.0	0.0.0.0

PC3 → PC1
PC3 → R5 → (212.1.2.65) R2 → (212.1.1.129) R1 → PC1
Para este último traxecto non é necesario engadir ningunha ruta adicional

Apartado c:

En R3 é necesario configurar unha nova dirección IP na interface da subrede F para que PC2 poida utilizar esa interface coma ruta por defecto hacia o resto das máquinas. Por exemplo, engadiríamos a dirección 212.1.4.1 á interface que ten a dirección 212.1.2.129
Ademais habería que modificar a ruta en R3:

Router	Destino	Máscara	Gateway
R3	212.1.4.0	255.255.255.0	0.0.0.0

Apartado d:

Subrede	Dirección de rede	Dirección de broadcast	Máscara		Rango IP's
Dpto 1	212.1.3.128	212.1.3.191	\26	255.255.255.192	212.1.3.129 – 212.1.3.190
Dpto 2	212.1.3.0	212.1.3.127	\25	255.255.255.128	212.1.3.1 – 212.1.3.126
Dpto 3	212.1.3.192	212.1.3.223	\27	255.255.255.224	212.1.3.193 – 212.1.3.222

Apartado e:

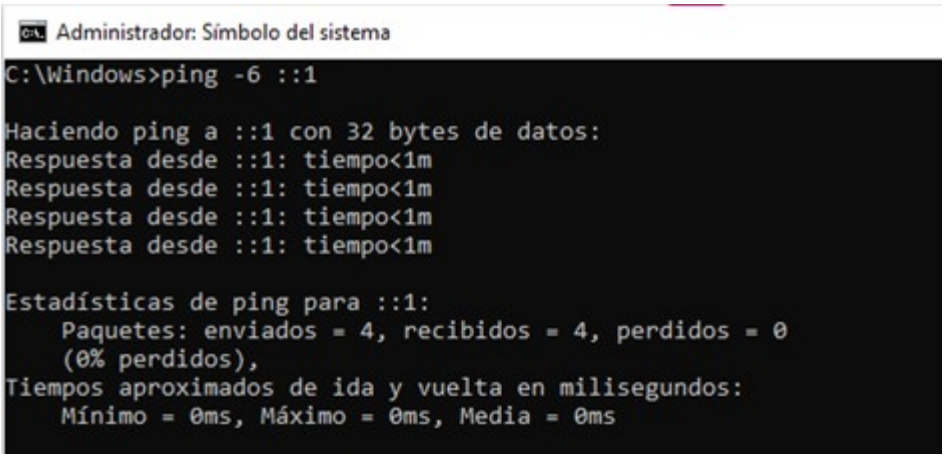
212.1.1.0 /27

Apartado f:

2001:0DB8:0000:0000:0000:0000:1428:57ab

Apartado g:

Trátase dunha proba de conectividade á propia interface de rede no protocolo IPv6. Dado cos sistemas operativos actuais implementan esta versión do protocolo, a resposta debe ser como a da imaxe.
::1 o 0::1 é o que se coñece como dirección de loopback (neste caso en IPv6)



Exercicio 4 da Opción A:

Unha posible solución sería:

```
/**********   Apartado e  **********/


#include <stdio.h>

//configuracion en compilacion
#define NUM_RES 5  //numero maximo de resistencias


//definicions de variables globales

int Resistencias[NUM_RES];
float Resultado = 0;
int NumeroElementos = 0;
char TipoOperacion = 's';     //s serie, p paralelo


//declaracion de funcions

int RecolleDatos(void);       //Recolle os datos das resistencias
float CalculaSerie(void);     // Calcula serie
float CalculaParalelo(void);  //calcula paralelo
void MostraResultado(void);   // mostra o resultado


//PARA DEBUG
#define DEBUG 0               //directiva compilacion debub



/**********   Apartado d  **********/

int main()
{

      int ok = 0; //variable de control
      char seguir;      //variable tipo caracter

      for (;;)    // bucle infinito salvo break
      {
            ok = RecolleDatos ();    //chamada á función RecolleDatos
            if (ok)                          //número elementos ok, operacion ok.
            {
                  if (TipoOperacion == 's')
                  {
                    printf ("Calculando asociación serie\r\n");
                    Resultado = CalculaSerie();  //chamada a la función CalculaSerie
                  }
                  else if (TipoOperacion == 'p')
                  {

                        printf ("Calculado paralelo\r\n");
                        Resultado = CalculaParalelo ();//chamada á función
CalculaParalelo

                  }
                  else   //  if (TipoOperacion != 's' && TipoOperacion != 'p')
                  {
                        printf ("Tipo de operación incorrecto. Adios\r\n");
                        break;
                  }   //Fin
```

```
                MostraResultado();//chamada a función MostraResultado

            }
            else   // Numero elementos non OK
            {
                printf ("Numero de elementos NON OK!\r\n");

            }  //FIN datos OK

            printf("Pulse ENTER para continuar o q para terminar\r\n");
            getchar();
            scanf("%c", &seguir);

            if (seguir == 'q')
            {
                break;
            }
    } // FIN for infinito

      printf("Adios");
      return 0;

} //FIN main



/**** Apartado a    *****/



int RecolleDatos(void)
{

      int ok = 0;
      int i;

      printf ("Número de elementos: ");
      scanf ("%d", &NumeroElementos);

      if ((NumeroElementos > 0) && (NumeroElementos < (NUM_RES + 1)))
      {

            ok = 1;

            for (i = 0; i < NumeroElementos; i++)
            {
                printf ("Elemento %d: ", i + 1);
                scanf ("%d", &Resistencias[i]);
            }

            printf ("Asociacion s serie, p paralelo: ");
            getchar ();
            scanf ("%c", &TipoOperacion);
      }

#if DEBUG //procesado condicional si DEBUG 1
      for (i = 0; i < 5; i++)
      {
            printf ("Elemento [%d,%d] %d: ", (i + 1), NumeroElementos,
            Resistencias[i]);
      }
#endif  //FIN debug

      return ok;
}
```

```
/**** Apartado b *******/



float CalculaSerie(void)
{
	float resultado = 0;
	int i;

	for (i = 0; i < NumeroElementos; i++)
	{
		resultado += Resistencias[i];
	}
	return resultado;
}



/**** Apartado d *******/


void MostraResultado(void)
{
	float resultadoK = 0;	//Resultado en Kohm

	resultadoK = Resultado / 1000;
	printf ("Resultado %f kOhm: ", resultadoK);

}



/**** Apartado c *******/



float CalculaParalelo(void)
{
	float resultado   = 0;
	int resultadoCero = 0;  // 1 si algunha resistencia e cero
	int i;

	for (i = 0; i < NumeroElementos; i++)
	{
		if (Resistencias[i] == 0)
		{
			resultado = 0;
			resultadoCero = 1;
			break;
		}
		else
		{
			resultado += 1 / (float) Resistencias[i];
		}
	} //FIN bucle calculo

	if (resultadoCero == 0) //
	{
		resultado = 1 / resultado;
	}

	return resultado;

}
```

Exercicio 5 da Opción A:

Apartado a:

Filtro Paso banda composto por un filtro Paso alto (R1-C1) e un filtro Paso baixo (R2-C2) xa que $f_{C1} < f_{C2}$

Apartado b:

$$f_{C1} = \frac{1}{2\pi \cdot R_1 \cdot C_1} = 312\text{Hz} \;\; ; \;\; f_{C2} = \frac{1}{2\pi \cdot R_2 \cdot C_2} = 4080\text{Hz}$$

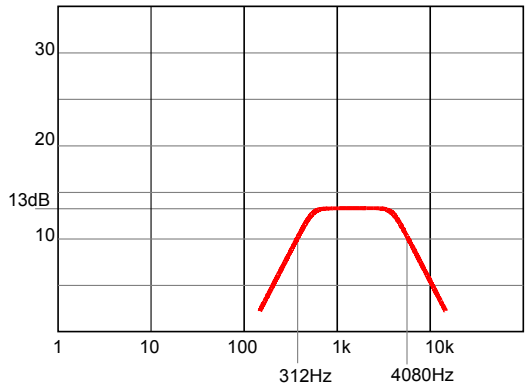
Apartado c:

$$BW = 3,77\text{kHz} \; ; \;\; f_C = 1128,25\text{kHz}$$

Apartado d:

$$AV_{dB} = 13\text{dB}$$

Apartado e:



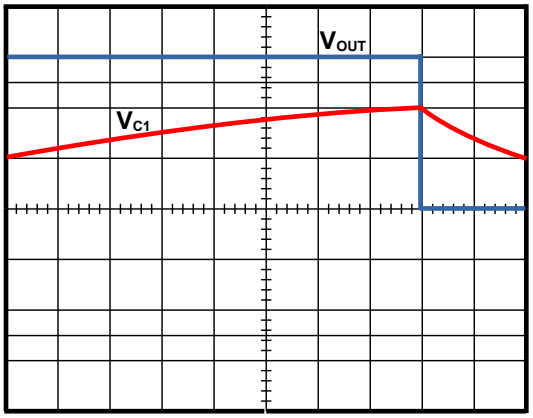
OPCIÓN B

Exercicio 1 da Opción B:

Apartado a:

$$T_H = (R1 + R2) \cdot C \cdot \ln 2 = (150k + 50k) \cdot 3 \cdot 10^{-6} \cdot 0,693 = 415ms \simeq 8DIV$$
$$T_L = R2 \cdot C \cdot \ln 2 = 50k \cdot 3 \cdot 10^{-6} \cdot 0,693 = 104ms \simeq 2DIV$$

Apartado b:



Canle A= 2V/div; Canle B= 2V/div; Time = 50 ms/div

Apartado c:

$$f = \frac{1}{T_H + T_L} = \frac{1}{415 + 104} = 1,92Hz \simeq 2Hz$$

Apartado d:

Para conseguir un ciclo de traballo do 50%, débese fixar R1=R2 e conectar un diodo rectificador en paralelo con R2 polarizado directamente, para que R2 no interveña no proceso de carga otro diodo en serie con R2 en inversa, para que R1 no influya na descarga. De esta forma conséguense uns tempos T_H e T_L, aproximadamente iguais.

Exercicio 2 da Opción B:

Apartado a:

Contador asíncrono binario ascendente de 4 bits

Apartado b:

Amplificador sumador inversor

Apartado c1:

$V_{OUTmin} = 0V; V_{OUTmax} = 9,375V$

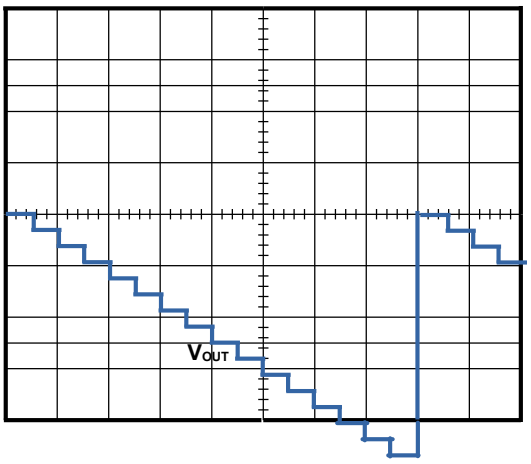
Apartado c2:

$f_{clock} = 200Hz; T = 5ms; 50ms \Rightarrow 10impulsos = 1010; operando\ V_{OUT} = -6,25V$

Apartado c3:

Valor absoluto da resolución: 0,125 · 5 = 0,625

Apartado d:



Canle A= 2V/div; Time = 10 ms/div

Ejercicio 3 da Opción B:

Apartado a:

I1	I2	I3	I4	MODO DE FUNCIONAMENTO
1	1	1	0	(1) - Carga paralelo, dato 10000000
1	1	0	1	(2) - Desprazamento esquerda
1	0	1	1	(3) - Desprazamento dereita
0	1	1	1	(4) - Pausa

I1, I2, I3 e I4 correspóndense coas entradas do codificador e xeneran o código correspondente de 2 bits que selecciona o modo de traballo dos rexistros.
As conexións entre as saídas e as entradas serie dos dous bloques, permiten desprazar 8 bits de forma cíclica e bidireccional á frecuencia definida polo reloxo de entrada.

Apartado b:

CLK	I1	I2	I3	I4	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
1	1	0	1	1	1	0	0	0	0	0	0	0
2	1	0	1	1	0	1	0	0	0	0	0	0
3	1	0	1	1	0	0	1	0	0	0	0	0
4	1	0	1	1	0	0	0	1	0	0	0	0
5	1	0	1	1	0	0	0	0	1	0	0	0
6	1	0	1	1	0	0	0	0	0	1	0	0
7	1	0	1	1	0	0	0	0	0	0	1	0
8	1	0	1	1	0	0	0	0	0	0	0	1
9	1	0	1	1	1	0	0	0	0	0	0	0
10	1	0	1	1	0	1	0	0	0	0	0	0

Apartado c:

Código contador en anel de 8 bits

Exercicio 4 da Opción B:

Apartado a:

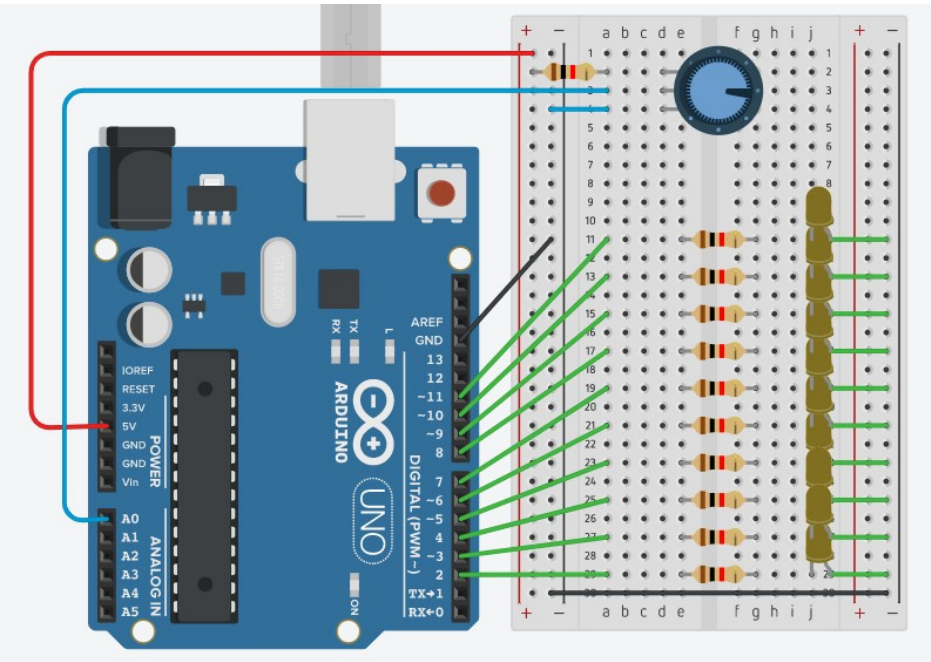
```
const int analogPin = A0; // Definir pin de entrada analóxica para conexión de sensor de entrada
const int ledCuenta = 10; // Número de LED de saída
int ledPins[ ] = {
2, 3, 4, 5, 6, 7, 8, 9, 10, 11
}; // Matriz de número de pin para conexión do LED de saída
void setup() {
// Bucle para configurar a matriz de pins como saídas para os LED
for (int Ledactivo = 0; Ledactivo < ledCuenta; Ledactivo++) {
pinMode(ledPins[Ledactivo], OUTPUT);
}
//
}
void loop() {
// Lectura do sensor o potenciómetro
int sensorReading = analogRead(analogPin);
// Equivalencia entre rango de sensor e rango de número do LED
int lednivel = map(sensorReading, 0, 1023, 0, ledCuenta);
// Bucle ON/OFF array de LED de saída
for (int Ledactivo = 0; Ledactivo < ledCuenta; Ledactivo++) {
//
// Activar pins de saída de valor inferior ao nivel do LED
if (Ledactivo < lednivel) {
digitalWrite(ledPins[Ledactivo], HIGH);
}
// Desactivar pins de saída de valor superior ao nivel do LED
else {
digitalWrite(ledPins[Ledactivo], LOW);
}
//
}
}
```

Apartado b:

```
void setup() {
// Inicializar comunicación serie
Serial.begin(9600);
}
// void loop() {
// Lectura da entrada analóxica en pin A1;
int sensorValue = analogRead(A1);
// Convertir lectura analóxica (0 - 1023) a voltaxe (0 - 5V)
  float voltage =  sensorValue * (5.0 / 1023.0);
// Presentar o valor da voltaxe por monitor serie
Serial.println(voltage);
```

Apartado c:

Unha posible solución soamente para o programa do apartado a, sería:



Exercicio 5 da Opción B:

Apartado b:

Unha posible solución sería:

R1		
Rede/mask	Gateway	Descripción
B	0.0.0.0	DC
192.168.5.0/24	0.0.0.0	DC
192.168.7.0/24	0.0.0.0	DC
A	200.1.17.20	RED BAJO R6
0.0.0.0	192.168.7.2	DEFAULT A R2
R2		
Rede/mask	Gateway	Descripción
C	0.0.0.0	DC
192.168.6.0/24	0.0.0.0	DC
192.168.7.0/24	0.0.0.0	DC
E	200.1.28.14	RED BAJO R4
F	200.1.28.1	RED BAJO R5
0.0.0.0	192.168.6.1	DEFAULT A R3
R3		
Rede/mask	Gateway	Descripción
D	0.0.0.0	DC
192.168.5.0/24	0.0.0.0	DC
192.168.6.0/24	0.0.0.0	DC
0.0.0.0	192.168.5.2	DEFAULT A R1
R4		
Rede/mask	Gateway	Descripción
C	0.0.0.0	DC
E	0.0.0.0	DC
0.0.0.0	200.1.28.2	DEFAULT A R2
R5		
Rede/mask	Gateway	Descripción
C	0.0.0.0	DC
F	0.0.0.0	DC
0.0.0.0	200.1.28.2	DEFAULT A R2
R6		
Red/mask	Gateway	Descripción
A	0.0.0.0	DC
B	0.0.0.0	DC
0.0.0.0	200.1.17.2	DEFAULT A R1

Apartado a:

REDE LAN	Dirección de rede	Dirección de broadcast	Máscara de rede	
A	200.1.17.32	200.1.17.63	/27	255.255.255.224
B	200.1.17.0	200.1.17.31	/27	255.255.255.224
C	200.1.28.0	200.1.28.15	/28	255.255.255.240
D	200.1.20.160	200.1.20.175	/28	255.255.255.240
E	200.1.28.248	200.1.28.255	/29	255.255.255.248
F	200.1.28.16	200.1.28.31	/28	255.255.255.240

Apartado c:

172.31.255.240 – 172.31.255.255

Apartado d:

2001:0DB8:0000:0000:0000:0000:1428:57ab

Apartado e:

Se trata dunha proba de conectividade á propia interfaz de rede no protocolo IPv6. Dado que os sistemas operativos actuales implementan esta versión do protocolo, a resposta debe ser como a da imaxe.
::1 o 0::1 é o que se coñece como dirección de loopback (neste caso en IPv6)