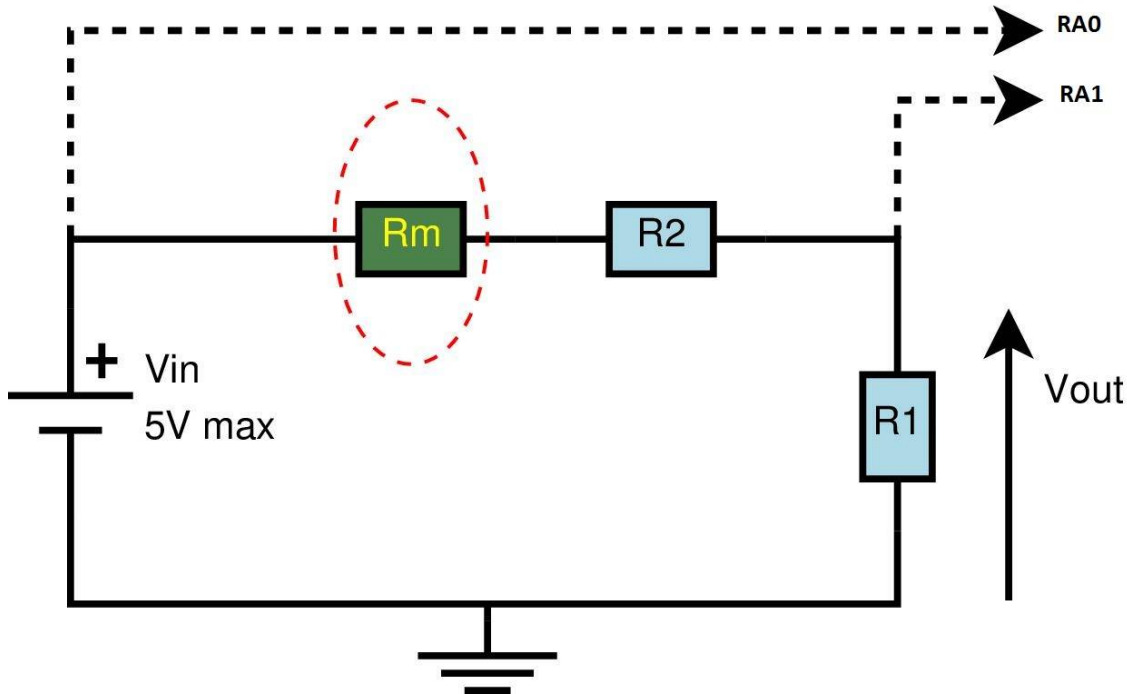




Etude de la cellule de mesure de la résistance (pont diviseur de la tension)



- V_{in} : Tension d'entrée égale 5V (valeur maximale)
- R_1, R_2 : Résistances du pont diviseur
- R_m : La **résistance** à mesurer par le **microcontrôleur**
- RA_0, RA_1 : Broches du microcontrôleur 16F877

Le principe de mesure est très simple !! On connaissant les paramètres : V_{in} , V_{out} , R_1 , R_2 on peut déterminer la valeur de la résistance R_m .

- R_1, R_2 : Les valeurs des résistances du pont sont définies par rapport à la dynamique de la tension de sortie et la valeur de la résistance à mesurer.
- V_{in}, V_{out} : On peut déterminer ses tensions en utilisant deux entrées analogiques du microcontrôleur (RA_0 et RA_1). Le micro dispose de quatre entrées analogiques avec une résolution de 10 bits (pas de quantification q d'environ 4.9 mV pour une alimentation de 5V).



Calcul de la résistance R_m :

$$V_s = V_{in} \frac{R_1}{R_m + R_1 + R_2}$$

$$\frac{V_s}{V_{in} \cdot R_1} = \frac{1}{R_m + R_1 + R_2}$$

$$R_m = R_1 \cdot \left[\frac{V_{in}}{V_s} - 1 \right] - R_2$$

Choix des résistances R_1 et R_2 :

Le choix de la résistance R_1 et R_2 dépend de la valeur de résistance à mesurer [R_{m_min} , R_{m_max}] et de la dynamique de la tension de sortie [V_{smin} , V_{smax}].

- La tension V_{smax} correspond à $R_m = 0$! $V_{smax} = V_{in} \cdot R_1 / (R_1 + R_2 + 0)$. En pratique on choisi R_1 et R_2 de telle sorte que la tension V_{smax} soit plus proche de la valeur maximale de l'ADC (5V).
- La tension V_{smin} correspond à la valeur maximale de R_m (R_{m_max}) : $V_{smin} = V_{in} \cdot R_1 / (R_1 + R_2 + R_{m_max})$; en Pratique on choisi la valeur des résistances pour avoir une tension V_{smin} plus proche de zero.

L'intérêt d'avoir une tension $V_{smax} \sim 5V$ et $V_{smin} \sim 0V$ est d'exploiter la dynamique totale de l'ADC du microcontrôleur afin de gagner en précision sur la mesure de la résistance R_m !!

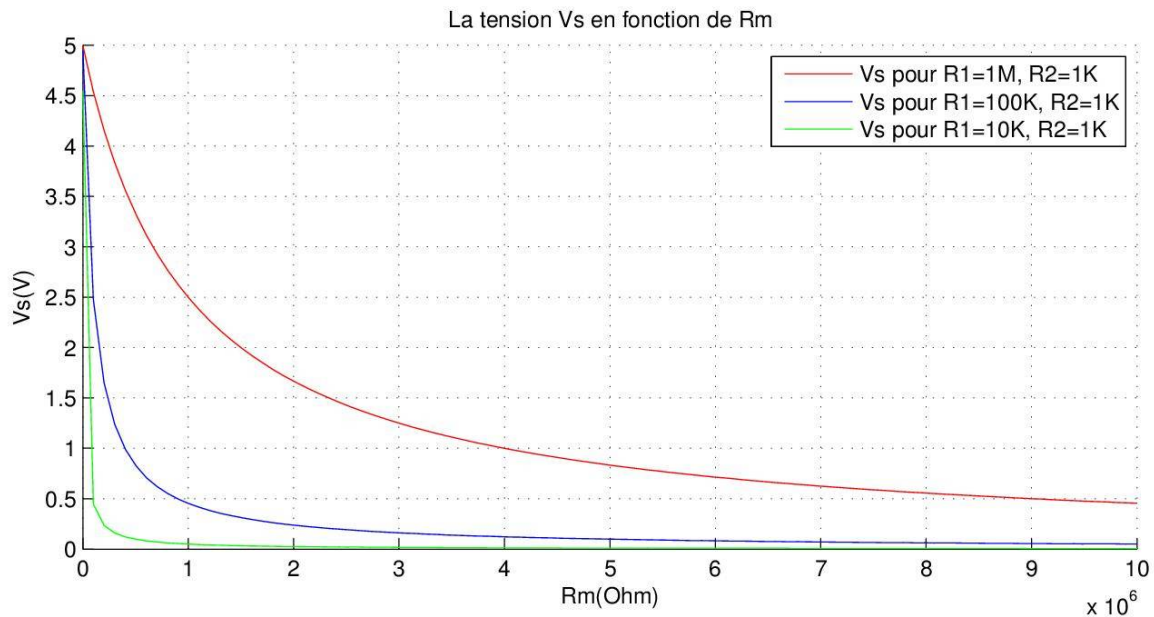
La tension V_s en fonction de la résistance R_m :

Paramètres de simulation :

- $V_{in} = 5v$



- $R_1 = 1\text{MR}$, 100KR ou 10KR
- R_2 fixe = 1KR
- R_m variée entre 0 et 10MR



On constate que pour mesurer une résistance faible ($<1\text{MR}$), la résistance R_1 doit être faible ($<100\text{K}$) (courbe vert et bleu).

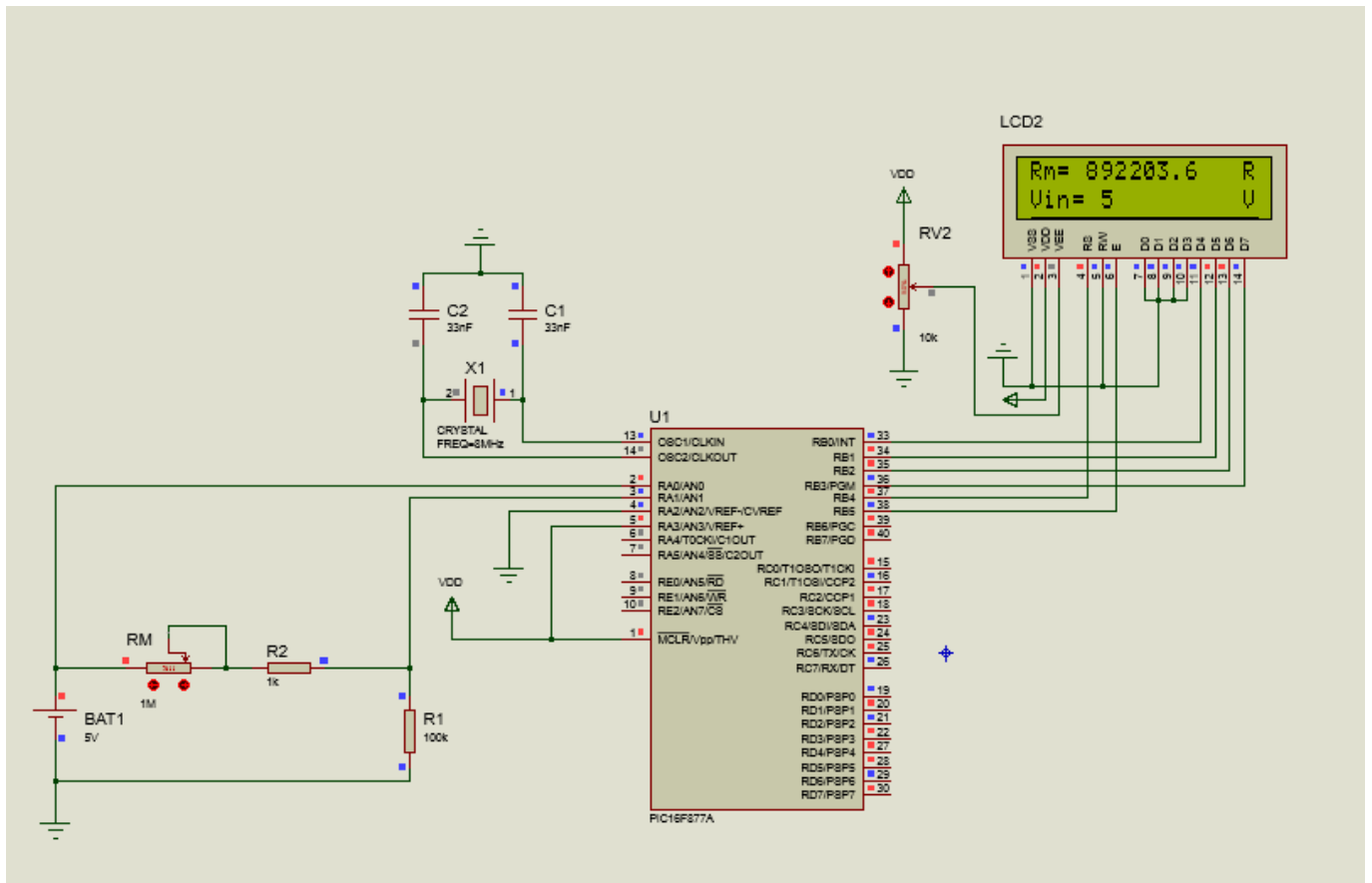
En revanche si on veut calculer une résistance de l'ordre de quelques MR, on doit utiliser $R_1 > 1\text{MR}$

Dans notre projet, la valeur de la résistance R_m variée entre 0 et 1MR d'où le choix de $R_1=100\text{K}$ et $R_2=1\text{K}$ ($V_{s\text{max}}=4.9505\text{V}$, $V_{s\text{min}}= 0.0495\text{V}$).

Schéma électronique du projet mesure



de la résistance (ISIS):





```
67
68 //Affichage sur LCD
69 Lcd_Out(1,1, Resohm); // Rm=
70 Lcd_Out(1,16, "R"); // R
71 Lcd_Out(2,1, VoltStr); // Vin=
72 Lcd_Out(2,16, "V"); // V
73
74 Lcd_Out(1,5,RmStr);
75 Lcd_Out(2,6,VinStr);
76
77 delay_ms(500);
78
79 Lcd_Cmd(_LCD_CLEAR);
80
81
```

Code MikroC :

```
// Connexions LCD
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;

const int Taille_LCD =16 ;

// Variables tensions (vin, vout)
unsigned int Vout_ADC;
float Vout_tension;
```



```
unsigned int    Vin_ADC;
float           Vin_tension;

// Paramètres de la cellule de mesure
float           Res1 = 100e3; // R1 =100k
float           Res2= 1e3;    // R2= 1k
float           Rm;
float           Vin = 5.0;
float           ADC_Ref = 5.0;

//Variables de l'affichage (résistance & tension Vin)
char            RmStr[Taille_LCD] ;
char            VinStr[Taille_LCD];
char            VoltStr[]="Vin=";
char            Resohm[]="Rm=";

void main()
{

    TRISA = 0xFF;           // PORTA en entrée
    TRISC = 0x00;          // PORTC en sortie
    TRISB = 0x00;          // PORTB en sortie
    TRISD = 0x00;          // PORTD en sortie

    ADC_Init();            // Initialisation ADC
    Lcd_Init();            // Initialisation LCD
    Lcd_Cmd(_LCD_CLEAR);  // Effacer LCD
    Lcd_Cmd(_LCD_CURSOR_OFF); // Désactiver le curseur

do
{
    //Lecture ADC (0 et 1)

    Vin_ADC =ADC_read(0);
    Vout_ADC =ADC_read(1);

    //Conversion de la valeur en tension
    Vin_tension = (float)Vin_ADC *ADC_Ref/1023.0;
    Vout_tension = (float)Vout_ADC *ADC_Ref/1023.0;

    // Calcul de la valeur de la résistance
    Rm = Res1*(Vin_tension/Vout_tension - 1) - Res2;
    if(Rm<0.0) Rm=0.0;
```



```
// Conversion de la valeur de Rm en chaîne de caractère
FloatToStr( Rm, RmStr );
FloatToStr( Vin_tension, VinStr );

//Affichage sur LCD
Lcd_Cmd(_LCD_TURN_ON);
delay_ms(10);
Lcd_Out(1,5,RmStr);
Lcd_Out(1,1, Resohm);
Lcd_Out(1,16, "R");
Lcd_Out(2,6,VinStr);
Lcd_Out(2,1, VoltStr);
Lcd_Out(2,16, "V");
delay_ms(1000);
Lcd_Cmd(_LCD_CLEAR);

} while(1);
}
```

Télécharger gratuitement le fichier du projet Ohmmètre numérique (Schéma ISIS + Code MikroC) :

• **Fichier Projet Ohmmètre Numérique**



Projet électronique : Ohmmètre numérique (PIC16F877A)

Revenir au sommaire des projets électronique

Click to rate this post!

[Total: 2 Average: 5]