

## Afficheur de tensions de module d'une batterie au NiMh

## Les batteries au NiMh

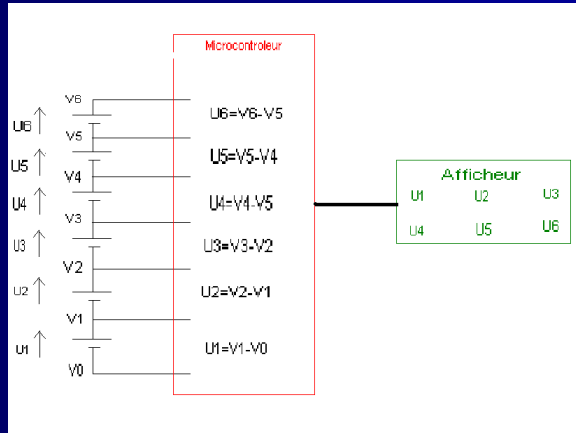


## Cahier des charge

- On devra afficher six tensions allant de 0,3 à 1,5 Volts
- La précision des mesures et de l'affichage devra être bonne.
- Utilisation du Microcontrolleur AT Mega 8535 et d'un écran LCD 16X4
- Alimentation 0-15Volts

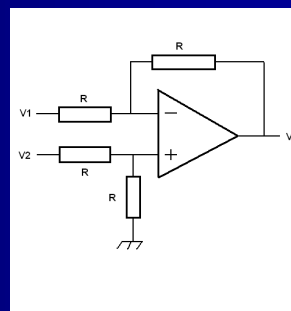
## Réalisation du projet

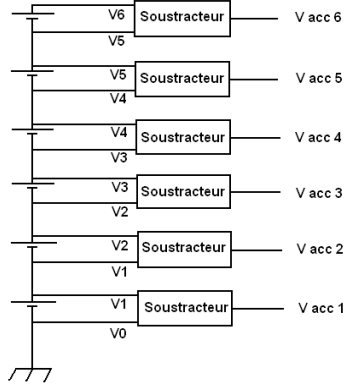
# Solution 1



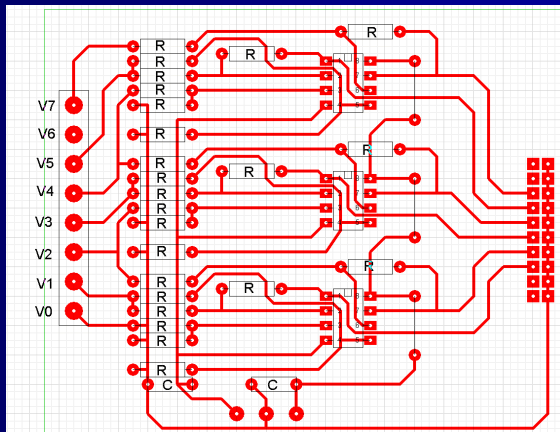
# Solution 2

- Le soustracteur

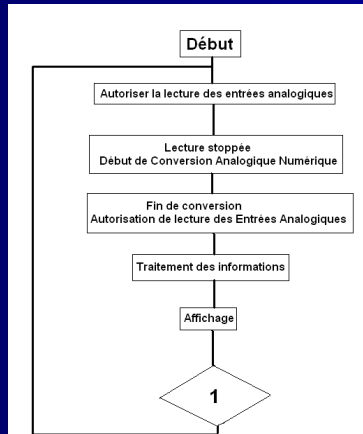




*Typom*



# Traitement et affichage des tensions



## La fonction « read\_adc »

- unsigned int read\_adc(unsigned char adc\_input)
- {
- ADMUX=adc-input|ADC\_VREF\_TYPE;
- ADCSRA|=0x40;
- while((ADCSRA & 0x10)==0)
- ADCSRA|=0x10;
- return ADCW;
- }

# ADMUX: Détermine l'entrée à lire et l'intervalle de tension de lecture

- -ADC\_VREF\_TYPE=0x60
- N° bit 7 6 5 4 3 2 1 0
- 0x60 0 1 1 0 0 0 0 0
  
- Bit 6-7: Indique que la lecture se fait sur 0, 5 volts.
  
- -ADC\_VREF\_TYPE | adc1= 0x60 = 0 x01= (011000001)b
- N° bit 7 6 5 4 3 2 1 0  
0 1 1 0 0 0 0 1
  
- Bit4-0: Numéro de la patte en lecture et type de lecture (simple ou différentiel) d'après le tableau suivant:

MUX4..0	Single Ended Input
00000	ADC0
00001	ADC1
00010	ADC2
00011	ADC3
00100	ADC4
00101	ADC5
00110	ADC6
00111	ADC7
01000	

# ADCSRA: Commande la lecture, la conversion

- -ADCSRA=0x84
- N°Bit 7 6 5 4 3 2 1 0 Bit 7: A 1 il autorise la lecture des entrées analogiques
- 0x84 1 0 0 0 0 1 0 0. Bit 2-0: Détermine le facteur de division K de la fréquence de l'horloge F pour donné la fréquence de lecture f.  $f=F/K$
- -ADCSRA=0x40
- N°Bit 7 6 5 4 3 2 1 0
- 0x40 0 1 0 0 0 0 0 0 Bit 6: A 1 il lance la conversion.
- -ADCSRA=0x10
- N°Bit 7 6 5 4 3 2 1 0 Bit 7: Est mise a 1 lorsque la conversion est Terminée
- 0x10 0 0 0 1 0 0 0 0

- unsigned int read\_adc(unsigned char adc\_input)  
« La fonction reçoit le numéro de l'entrée a lire et renvoie sa tensions convertie »
- {
- ADMUX=adc-input|ADC\_VREF\_TYPE;  
« Calibrage de l'entrée à 0-5V »
- ADCSRA|=0x40; « Lancement de la conversion »
- while((ADCSRA & 0x10)==0) « Tant que la conversion n'est pas finie »
- ADCSRA|=0x10; « Continuer la conversion »
- return ADCW; « Retourner la valeur convertie »
- }

# Explication de la fonction main

- unsigned int V1,V11,V111;
- char Tampon1;
- While(1)
- {
- V1=read\_adc(2);
- V11=V1\*0,004882125;
- V111=(V1\*0,004882125-V11)\*100;
- lcd\_goto(0,0);
- sprintf(tampon1, « V1=%d,%2dV »,V1,V11);
- lcd\_puts(tampon1);
- }

## Planning

Semaine	Travaux
1	Mise en place du cahier des charge.
2	Etude d'une première solution vers une deuxième solution
3-4-5-6	Conception de la carte
7-8-9-10	Programmation du microcontrolleur
11-12	Dossier



# Conclusion