

Rapport d'Électronique Analogique et Numérique

Construction d'un prototype de capteur de température

Valentine BUTTERATI - Mathys DEMARQUE - Thibault MORETTI -
Théo NGUYEN - Julien ROUMANILLE

Sommaire

Introduction.....	3
APP 1 : Grandeur Physique.....	4
APP 2 : Conditionner le signal électrique.....	7
APP 3 : Traitement numérique.....	10
APP 4 : Mémorisation et affichage.....	13
APP 5 : Signal horloge.....	16
Conclusion.....	19

Introduction

Le projet de ce semestre consiste en la création d'un montage qui permet d'afficher la température de la pièce par des composants électroniques, et sans carte Arduino. Le projet se décompose en 5 APP, au cours desquels nous devons nous familiariser avec un ou plusieurs composants et utiliser les ressources à notre disposition pour répondre à chaque problématique. Pour cela, nous devons travailler en équipe, repérer les informations importantes dans les documentations techniques, apprendre à lire et modifier un signal tout en mettant en pratique nos connaissances théoriques ainsi qu'analyser nos potentielles erreurs.

APP 1 : Grandeur Physique

Problématique : Acquérir et conditionner un signal.

Composants utilisés : AOP MCP602, capteur LM35

Objectifs : se familiariser avec les composants, choisir un type de circuit approprié par rapport à l'énoncé, monter le circuit et vérifier son fonctionnement

Choix du branchement de l'AOP

Type d'amplification

Objectif : trouver une amplification qui permette d'avoir une précision à l'unité de température

Amplification simple : plus de perturbations (jusqu'à 1%) mais montage plus simple

Amplification différentielle : amplification de la valeur qui nous intéresse uniquement, donc moins de perturbations mais montage et calculs plus complexes

On fait donc le choix d'une **amplification simple**, l'énoncé demande une précision à l'unité de température (donc à 0,1V près) ce qui est supérieur au 1% de perturbation.

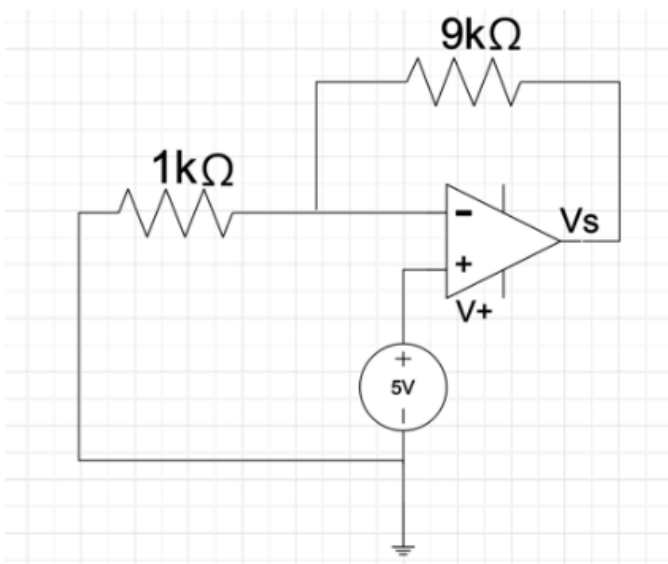
Compétence(s) mobilisée(s) :

- Calculer un TRMC pour évaluer le taux de perturbation de notre système

Type de montage

Objectif : trouver un montage qui permette d'avoir un facteur 10 entre la tension d'entrée (sortie du capteur) et la tension de sortie (tension à afficher)

Montage non inverseur

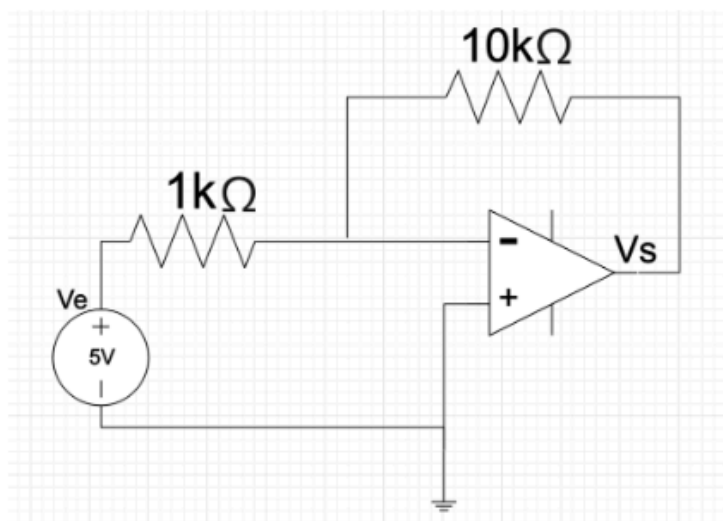


Au final,

$$\begin{aligned} -\frac{V_E}{R_1} &= \frac{V_E - V_S}{R_2} \\ V_E \left(\frac{1}{R_1} + \frac{1}{R_2} \right) - \frac{V_S}{R_2} &= 0 \\ \frac{V_S}{V_E} &= \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \times R_2 \end{aligned}$$

$$\frac{V_S}{V_E} = \left(\frac{R_2}{R_1} + 1 \right)$$

Montage inverseur



Au final,

$$\frac{V_E}{R_1} + \frac{V_S}{R_2} = 0$$

$$\frac{V_E}{R_1} = -\frac{V_S}{R_2}$$

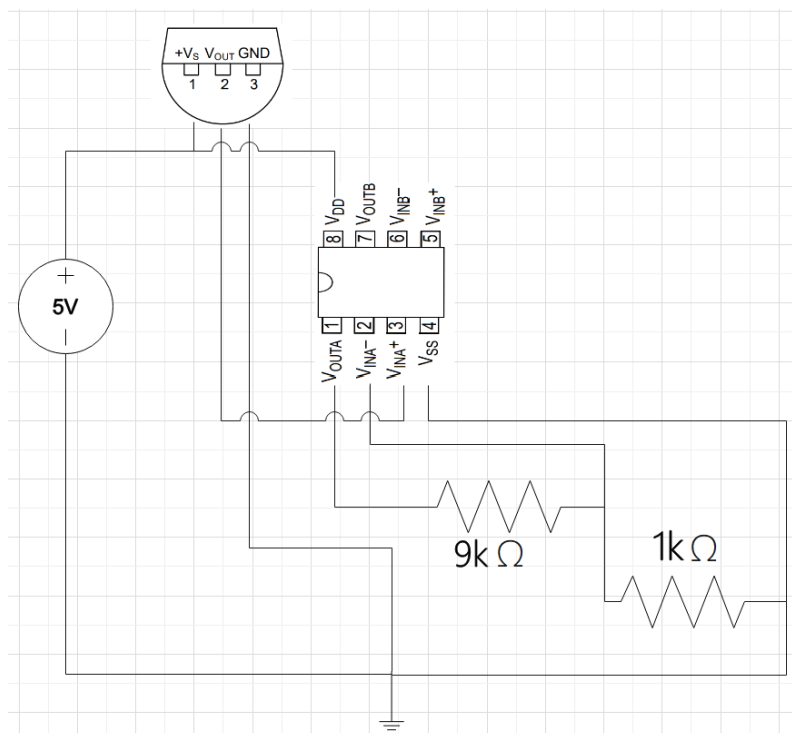
$$\frac{V_E}{V_S} = -\frac{R_1}{R_2}$$

Choix d'un montage non-inverseur pour éviter la présence d'un facteur négatif. On veut une amplification de 10, soit $\frac{R_2}{R_1} + 1 = 10$

Compétence(s) mobilisée(s) :

- Connaître les propriétés des AOP et de leur montages

Montage général



Compétence(s) mobilisée(s) :

- Lire et comprendre les datasheets

Conclusion

L'objectif de cet APP a été de trouver comment **amplifier par 10 un signal analogique** sortant du **LM35**, qui est le capteur de température.

Pour cela, nous avons réfléchi au branchement de notre amplificateur opérationnel, le **MCP602**. L'étude du **TRMC** nous a servi à évaluer la capacité de l'AOP à **rejeter ou atténuer les signaux présents en mode commun**, donc à choisir notre type d'amplification.

Ainsi, après avoir étudié les différents branchements possibles et les différentes caractéristiques de l'AOP, nous avons choisi de faire un **montage en non-inverseur simple avec un rapport entre les résistances de 9**.

Durant cet APP, nous avons appris à trouver les informations dont nous avons besoin dans une datasheet et à appliquer les connaissances des années précédentes.

APP 2 : Conditionner le signal électrique

Problématique : Convertir un signal analogique en numérique.

Composants utilisés : CAN ADC0832

Objectifs : comprendre le fonctionnement du CAN, interpréter les résultats en binaire, adapter notre utilisation du CAN à l'énoncé

Choix du mode

Il existe 2 types de modes :

- **RD** : "Reading" qui permet de convertir la tension analogique donnée en V_{in} , en sortie numérique binaire.
- **WR-RD** : "Writing et Reading" qui permet de stocker la valeur dans une mémoire.

On choisit le mode Reading puisque le stockage se fera à une étape différente du projet.

Conversion

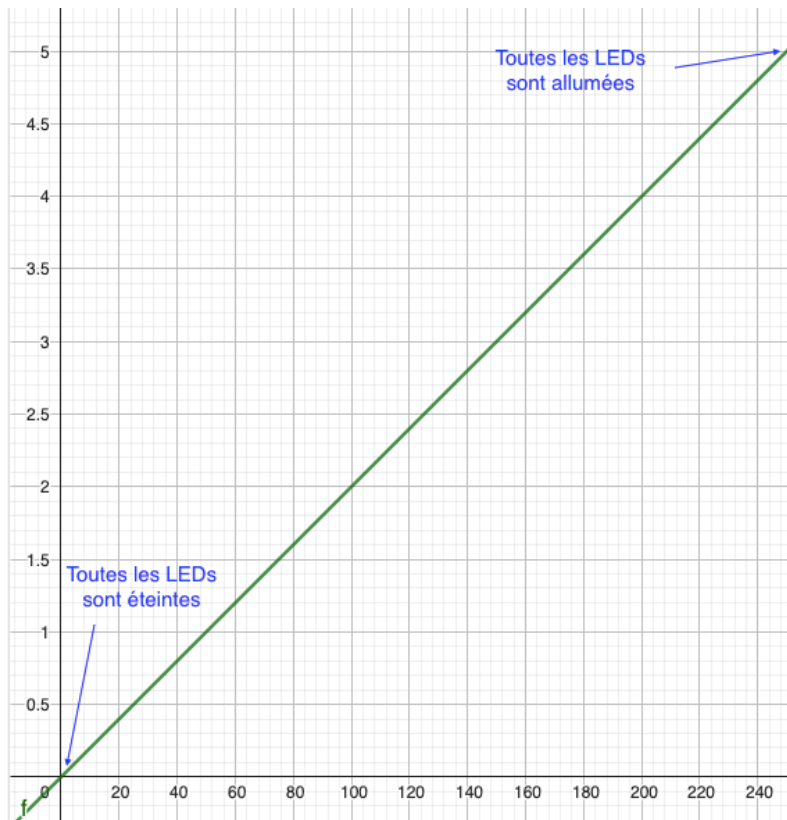
On calcule le quantum grâce aux différentes informations récupérées dans la datasheet :

- **Quantum** : $q = \frac{V_r}{2^n}$ avec V_r l'écart entre la tension d'entrée maximale et minimale (en V), n le nombre de bits du convertisseur.
- **Tension d'entrée max** : La température max reçue par le capteur est de 150°C donc le capteur a une sortie max de 150mV. Ainsi, la sortie max de l'ampli op correspond à 10x la tension de sortie du capteur, soit 15V. Or, $15V > V_{sat} = 5V$ donc $V_{max} = 5V$.
- **Tension d'entrée min** : En suivant le même raisonnement, on trouve $V_{min} = 0,02V$
- **Résolution du can** : 8 bits
- $V_r = V_{max} - V_{min} = 5 - 0.02 = 4.98V$

$$\Rightarrow \text{donc } q = \frac{4.98}{2^8} \approx 0,02V$$

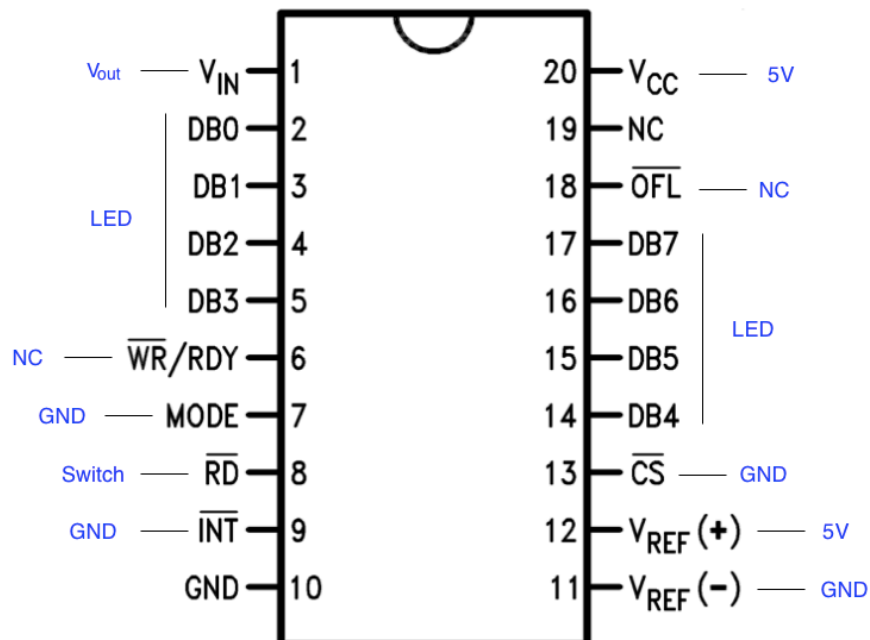
La conversion se fait alors sous forme d'une fonction linéaire $f(x) = qx = 0,02x$.

D'où notre résultat expérimental : toutes les LED sont éteintes (0 en binaire) lorsqu'on est à 0V et toutes les LED sont allumées (255 en binaire) lorsqu'on est à 5V.

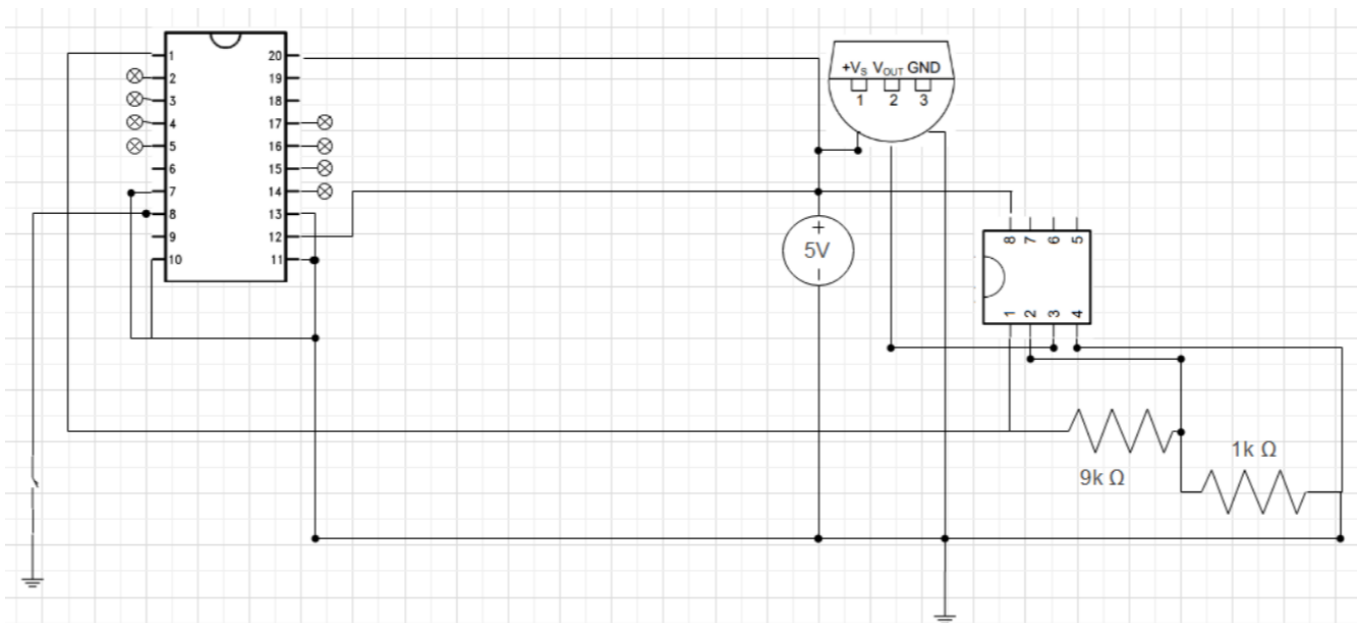


Branchement

Le switch déclenche un front descendant qui réactualise la conversion.



Montage général



Conclusion

Au final, notre CAN permet la conversion d'une donnée analogique (la tension de sortie de l'AOP) en une valeur numérique, ce qui va nous permettre plus tard de l'afficher. La réactualisation se fait sur le front descendant du switch connecté au pin RD.

On vérifie nos résultats en comparant le total des LED en binaire avec la tension mesurée au multimètre de la platine.

La difficulté principale a été de comprendre le rôle de chaque pin du CAN, mais le branchement s'est globalement passé sans accrocs.

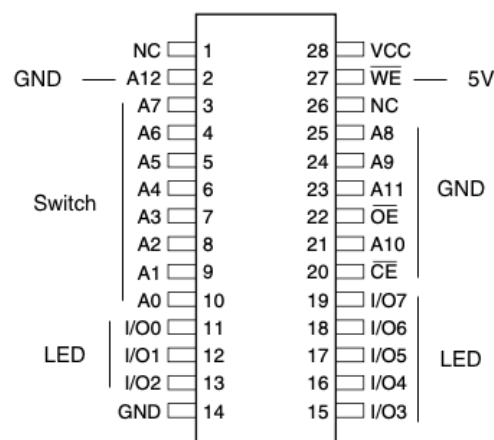
APP 3 : Traitement numérique

Problématique : Utiliser une EEPROM programmée et adapter la sortie CAN à l'entrée de l'EEPROM

Composants utilisés : EEPROM

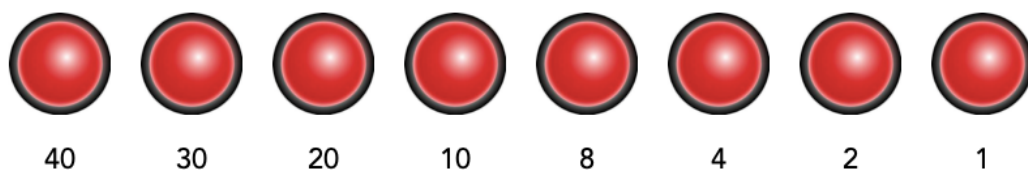
Objectifs : comprendre la différence entre adresses et données, comprendre la programmation et le branchement de l'EEPROM

Lecture des infos pré-programmées sur l'EEPROM



Branchement de l'EEPROM pour trouver le programme intégré

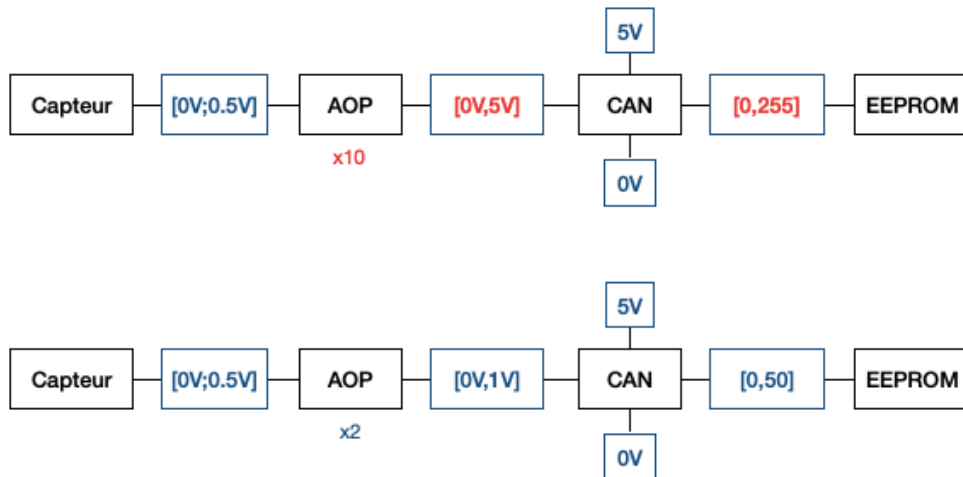
1. On branche à des switches les 8 premières adresses de l'EEPROM (puisque nos informations sont codées sur max 8 bits), le reste des adresses est branché à la masse. Les switches peuvent exprimer une valeur entre 0 et 2^7 .
2. On fait varier nos valeurs en observant les conséquences sur les LED \Rightarrow on remarque que pour les valeurs de 1 à 9, l'allumage des 4 premières LEDs est cohérent avec du binaire, mais pour les valeurs supérieures à 10 on voit que les 4 LEDs suivantes représentent les dizaines de 10 à 40.



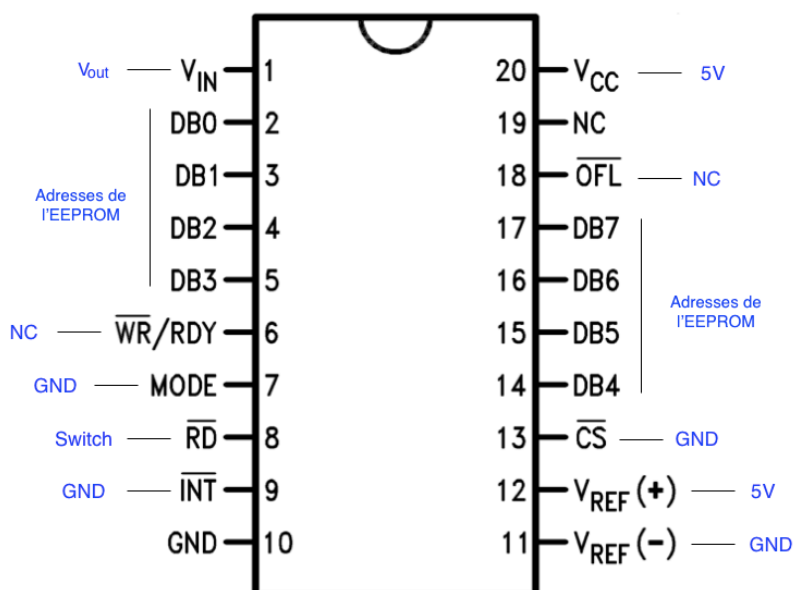
3. On cherche la valeur max atteignable, soit lorsque toutes les LED s'allument (overload). Ici il s'agit de $39 \cdot 2^5 < 39 < 2^6$, donc on en déduit que l'EEPROM est codé sur 6 bits.

Changement de gain

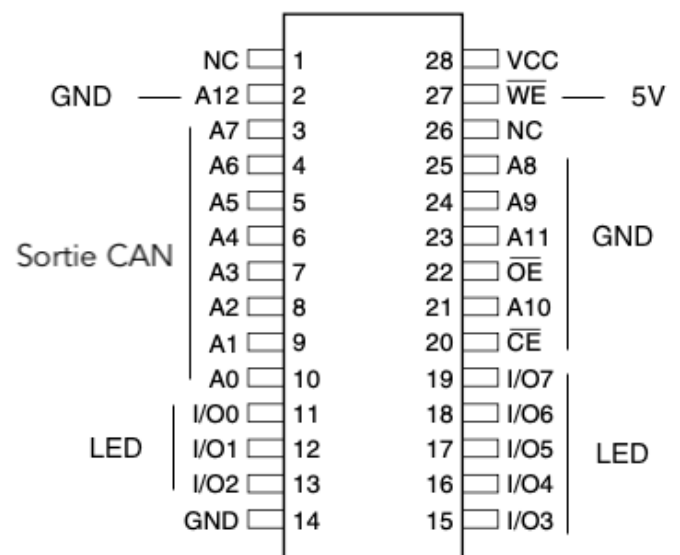
On cherche à réduire l'intervalle des valeurs qui rentrent dans l'EEPROM, pour cela on doit diminuer le gain de l'AOP. On remplace donc les résistances du branchement de l'AOP par deux résistances de valeurs égales (ici, $2k\Omega$) afin d'obtenir un gain de 2.



Branchements finaux

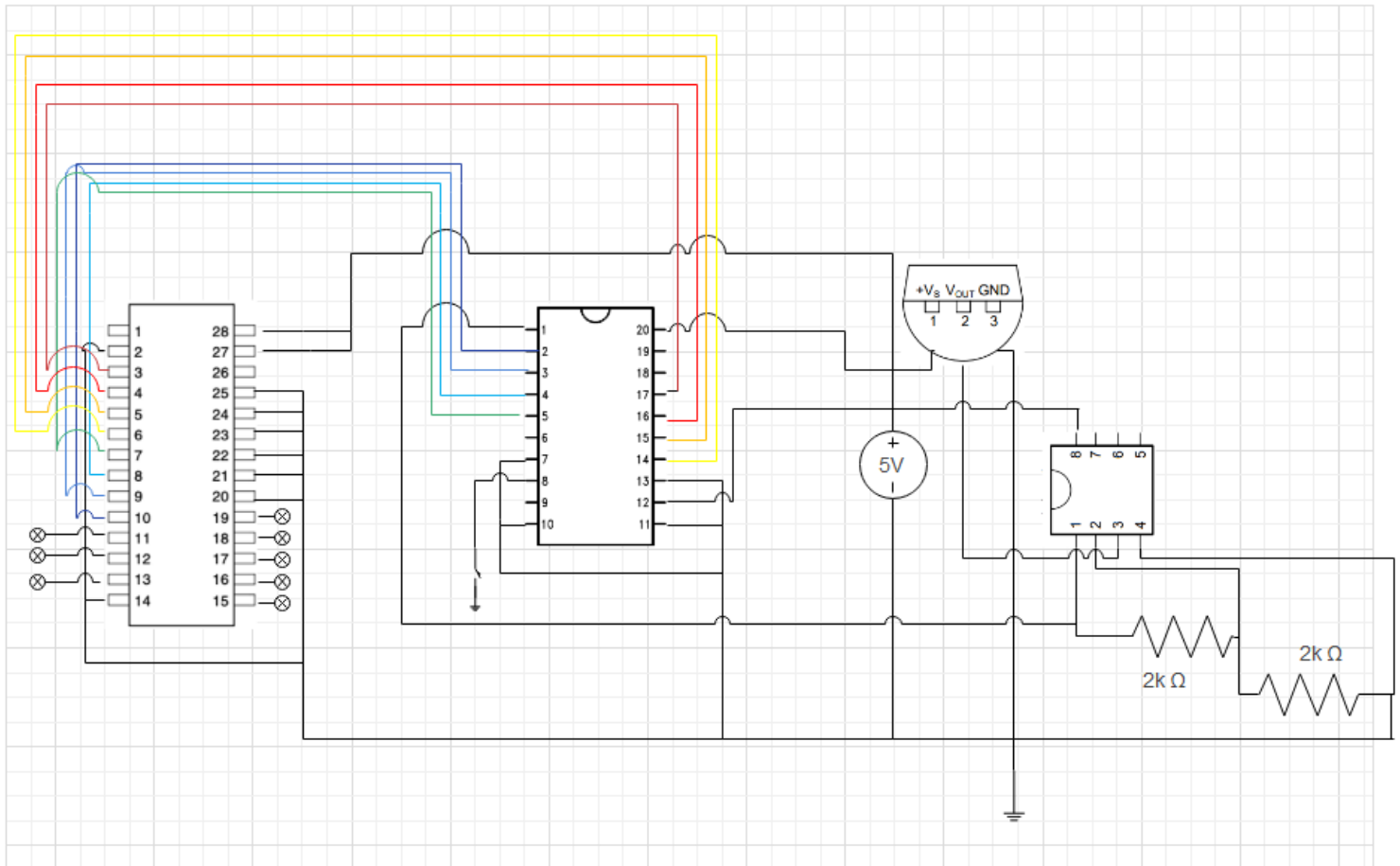


Branchement du CAN



Branchement de l'EEPROM

Montage général



Conclusion

Au final, notre EEPROM sert à stocker des données dans des adresses préalablement codées et va nous permettre de préparer l'affichage en séparant dizaines et unités.

Notre difficulté principale a été de se retrouver dans les branchements, nous avons donc commencer à raccourcir des fils et réarranger l'espace sur la breadboard.

APP 4 : Mémorisation et affichage

Problématique : Utiliser l'afficheur 7S & Codeur BCD, adapter la sortie de l'EEPROM à l'afficheur

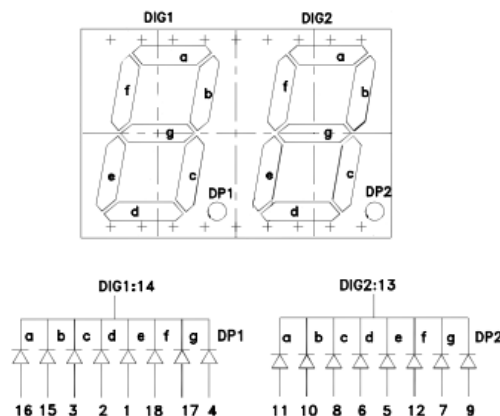
Composant utilisés : Afficheur 7 segments, Codeur CMOS BCD-to-7-segment

Objectifs : Comprendre le fonctionnement de l'afficheur et du codeur, l'utiliser avec l'EEPROM

Choix du branchement

Afficheur 7-segment

Après analyse de la datasheet, on remarque que 14 pins de l'afficheur (sur les 18 au total avec les 4 restantes connectées au GND) correspondent chacune à un segment. Le schéma ci-dessous montre les segments et les pins associés.



Choix des résistances R_{LED}

L'afficheur fait passer un courant de $30mA$ et a un potentiel V_{diode} de $0.7V$ dans chaque pins. Pour trouver la capacité des résistances adaptée à chaque pin de l'afficheur 7-segment, on a résolu cette équation :

$$R_{LED} = \frac{V_{cc} - V_{diode}}{I_{segment}} = \frac{5 - 0.7}{0.03} = 143.3\Omega$$

Avec :

V_{cc} le potentiel de notre platine de $5V$

V_{diode} le potentiel de la diode dans l'afficheur 7-segment

$I_{segment}$ le courant qui passe dans les diodes

C'est pourquoi nous avons choisis des résistances de 220Ω pour chacune des LED.

Compréhension du CMOS BCD-to-7-segment

Nous avons, comme pour les composants précédents, analysé la datasheet afin de comprendre l'utilité et comment se servir de ces compteurs.

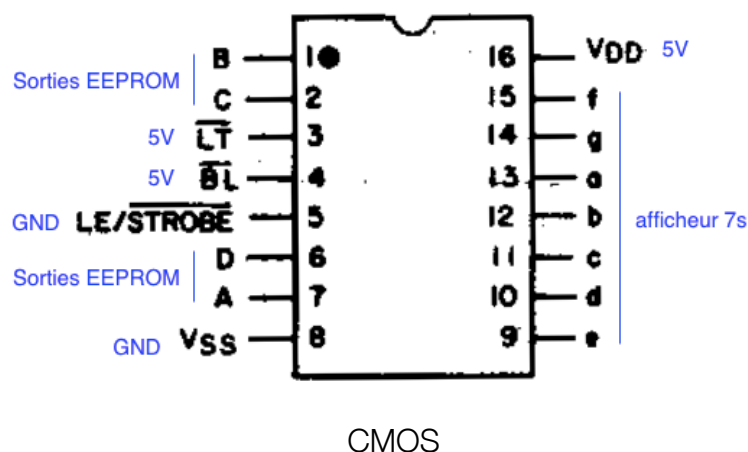
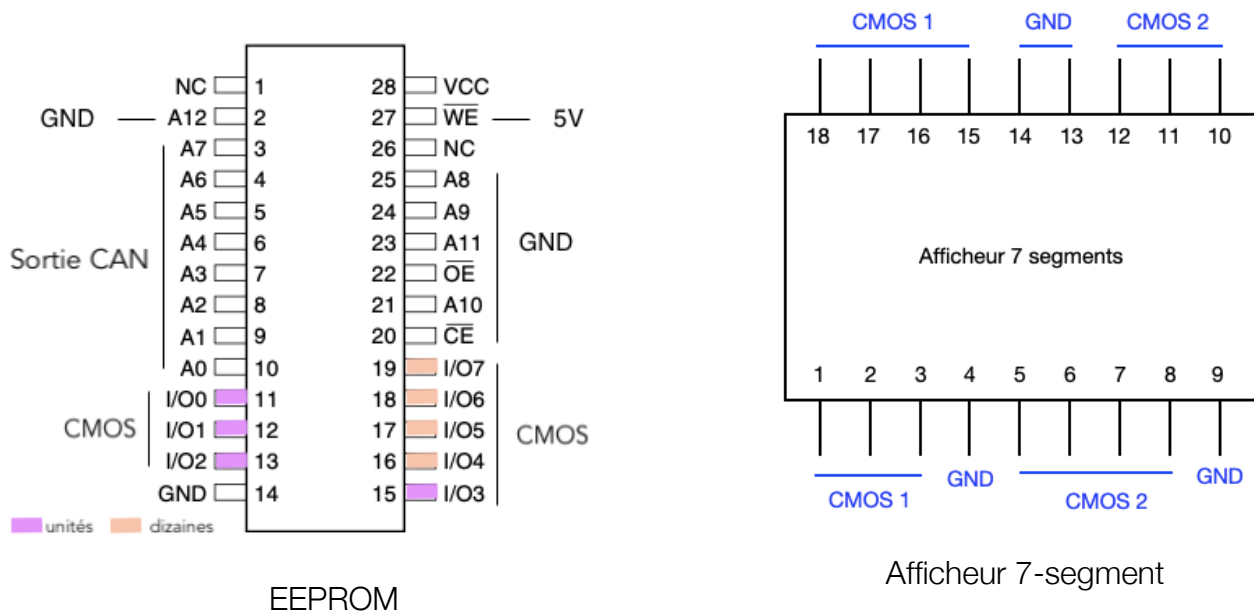
Ces compteurs vont nous permettre de pouvoir **convertir le signal binaire** que nous donne l'EEPROM en un signal affichable sur l'afficheur 7s.

D'après la datasheet, pour faire afficher des nombres par l'afficheur 7s, il faut que les pins LT et BL soient toujours à **l'état haut** et la pin LE/STROBE à **l'état bas**. On les branche donc respectivement à l'alimentation et au GND.

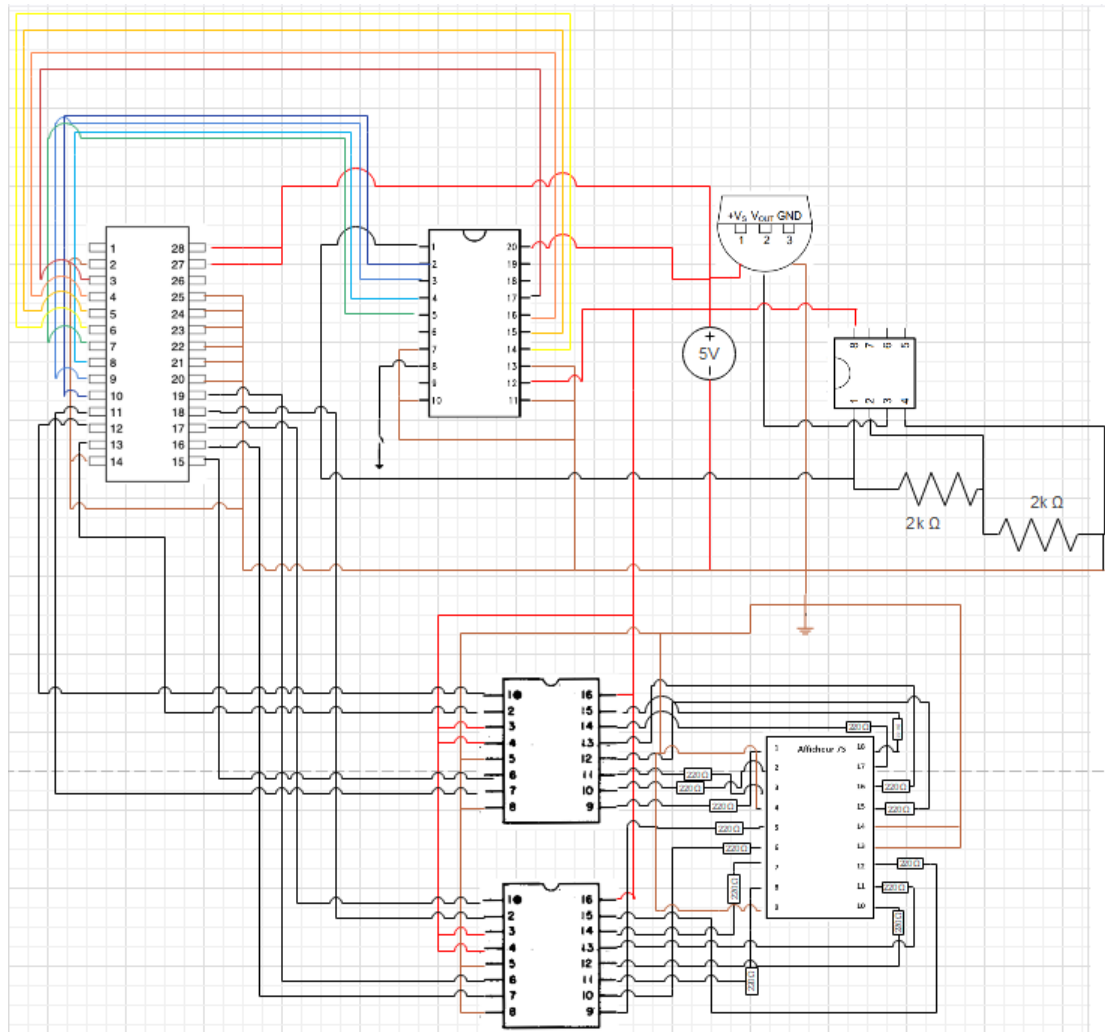
Comme l'EEPROM nous donne des chiffres binaires codés sur 8bits et que nos compteurs n'ont que quatre pins d'entrées, nous avons pris **deux compteurs**, un pour traiter les unités et l'autre pour les dizaines.

Nous avons ensuite branché l'EEPROM avec nos compteurs de façons à ce que les bons bits soient avec les bonnes pins du compteurs, donc le 2⁰ de l'EEPROM avec la pin du compteur qui va traiter cette information, et ce pour les dizaines également.

Branchements finaux



Montage Général



Conclusion

L'objectif de cet APP a été d'afficher la sortie de l'EEPROM (précédemment fait avec les LED de la platine) sur un afficheur 7-segment qui montre un **affichage décimal**.

Pour cela, nous avons dû brancher les CMOS-BCD à l'afficheur, et les pins de l'EEPROM (auparavant lié aux LED) aux 2 CMOS.

Pour vérifier nos branchements, nous avons d'abord mis une tension de 5V aux pins de chaque segment tout en observant si ceux-ci s'allumaient. Puis, nous avons branché les CMOS à l'afficheur et observé si **l'affichage correspondait à la valeur attendue**.

Les principales difficultés ont été le **branchement des résistances et du CMOS**. En effet, nous avons mis d'abord les résistances entre le CMOS et l'afficheur en **court-circuit**, ce qui dérégla le résultat affiché. En réglant ce problème, nous avons mis cette fois-ci les pins symétriques du CMOS en court-circuit qui a donc provoqué une **surchauffe au niveau des 2 composants**. Nous les avons donc remplacé en faisant cette fois-ci plus attention à nos branchements.

APP 5 : Signal horloge

Problématique : Comment utiliser le NE555 pour réaliser un signal rectangulaire.

Composant utilisés : NE555

Objectifs : Comprendre les différents fonctionnements possibles du NE555, implémenter l'horloge sur la platine en remplacement de l'interrupteur.

Choix du Branchement

Implémentation d'une horloge

Jusqu'à maintenant, nous utilisons un interrupteur de la platine relié au CAN pour rafraîchir la valeur affichée par l'afficheur 7-segments. Le but ici est de réaliser une horloge qui va automatiquement rafraîchir la valeur présente sur l'afficheur avec une période T choisie.

C'est pourquoi nous nous servons d'un NE555 qui va délivrer un signal créneau avec une période qui va dépendre du courant distribué dans les différentes pins, et donc des résistances utilisées. Cette connexion astable entraîne la charge et la décharge du condensateur C entre le niveau de tension de seuil ($\approx 0,67 \times VCC$) et le niveau de tension de déclenchement ($\approx 0,33 \times VCC$).

Calcul de R_A et R_B

On veut une résistance R_A et R_B afin d'obtenir une période T souhaitée de $0.5s$ pour un condensateur de $10\mu F$ donné.

$$\text{On a : } T = t_H + t_L = 0.693(R_A + 2R_B)C$$

Avec :

- $T = 0.5s$
- t_H le temps lorsque le signal est haut
- t_L le temps lorsque le signal est bas
- $C = 10\mu F$

$$\text{Donc } (R_A + 2R_B) = \frac{T}{(C \times 0.693)} = 72150\Omega = 72.15k\Omega$$

Nous avons utilisé 3 résistances : $R_A = 10k\Omega$, $R_B = 33k\Omega$, $R_L = 1k\Omega$

$$\text{On a donc } t_h = 0.693(R_A + R_B)C = 0.28s \text{ et } t_L = 0.693(R_B)C = 0.22s$$

Avec notre branchement final, on a donc un NE555 qui envoie un signal horloge avec une période de $0.5s$.

Pin diagram of the PIC16F877 microcontroller:

- Pin 1: GND
- Pin 2: TRIG
- Pin 3: PIN8 CAN
- Pin 4: RESET
- Pin 5: NC
- Pin 6: THRESH
- Pin 7: DISCH
- Pin 8: +5V

Conclusion

L'objectif de cet APP a été de remplacer l'interrupteur relié au CAN permettant de rafraîchir manuellement la valeur de l'afficheur 7-Segment par un signal horloge qui rafraîchira la valeur toutes les 0,5s.

Pour vérifier que le montage fonctionne, nous avons branché la sortie du NE555 à une LED qui devait donc clignoter lors du signal haut t_H sur une période de 0.5s.

Les principales difficultés de cet APP ont été de passer du schéma de la datasheet aux composants réels car les pins ne sont pas au même endroit. De plus, nous avons perdu du temps en échangeant le sens du condensateur et en devant changer du matériel défectueux (breadboard et afficheur). Nous avons quand même pu atteindre un résultat satisfaisant malgré une petite instabilité sur les valeurs affichées.

Conclusion

En conclusion, ces 5 APP nous ont permis de monter en compétence sur plusieurs domaines. Tout d'abord, nous avons tous pu approfondir nos connaissances techniques (câblage, conversion, traitement de données...) en lisant les datasheets et en participant à tour de rôle aux montages. De plus, nous avons réussi à nous organiser en équipe et nous répartir les tâches (montage, rédaction, dessin de schémas) d'une manière efficace. Même si certains d'entre nous ont plus fait la partie pratique que d'autres, nous avons fait en sorte que chaque personne soit à jour et actif dans le groupe. La partie la plus difficile des APP a été en général de nous adapter aux imprévus (composants ou matériel qui ne fonctionnaient pas, erreurs d'inattention en câblant, etc.) mais au final, cela ne nous a pas empêché d'avoir un résultat satisfaisant, qui remplissait la majorité des critères de départ.