

# Monitoring de l'alimentation du Maser

Ibrahim Khaled

## **Quelques explications :**

On veut réaliser un système qui s'assurent que le Maser du bureau de l'heure est toujours alimenté. Sachant que ce dernier peut être alimenté de deux manières :

- Secteur (avec convertisseur AC/DC)
- Batterie 26V

Via ce système, on pourra localiser suivant différents cas de figurent où se trouve la panne d'alimentation et envoyer un mail, sms aux responsables du bureau grâce au réseau local.

## **Montage :**

Le montage se présente comme suit :

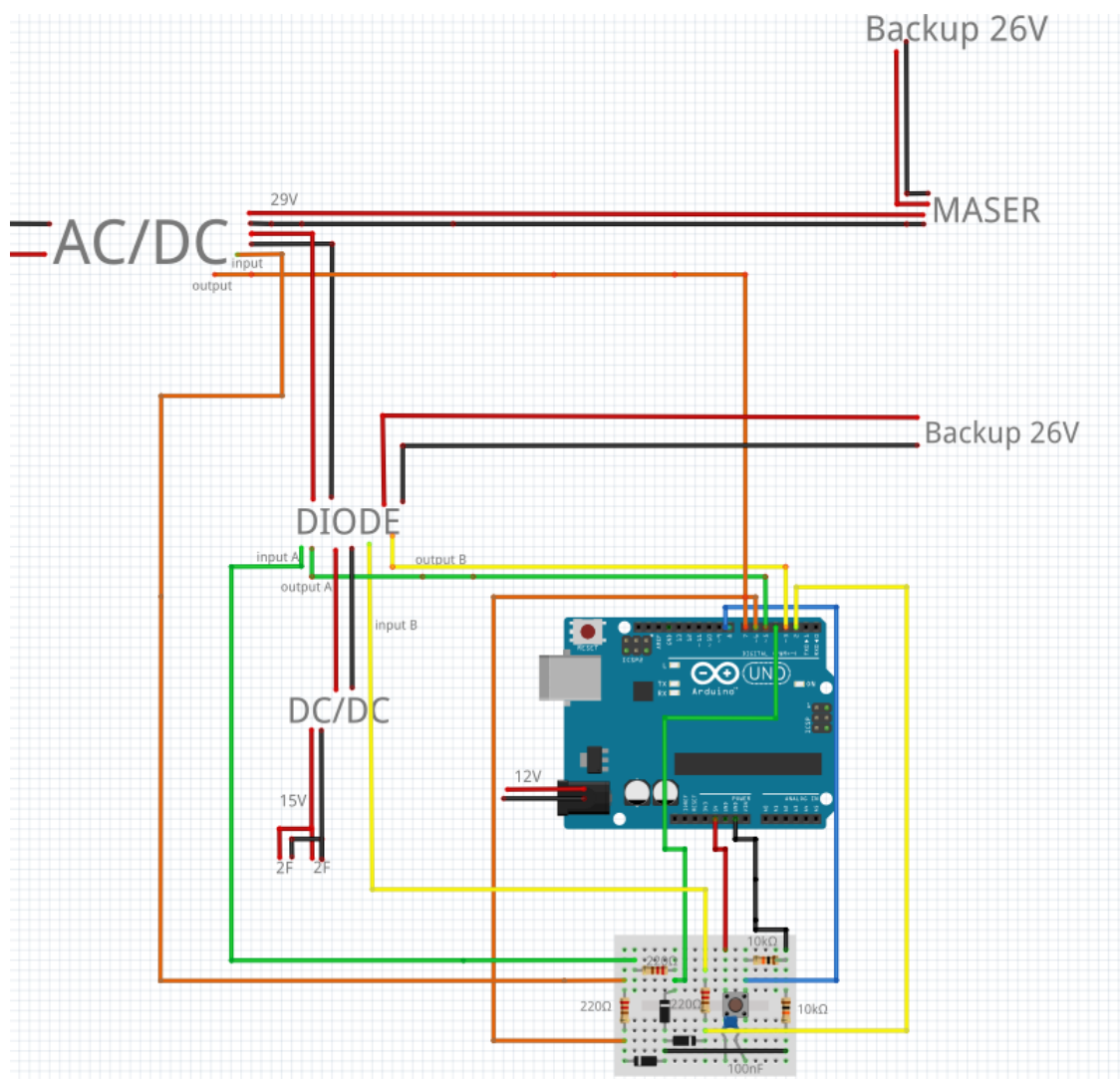


FIGURE 1 – Schéma descriptif

on y retrouve le convertisseur AC/DC pour l'alimentation du Maser, la diode qui joue l'échange de rôle d'alimentation entre le backup batterie et le secteur et enfin un convertisseur DC/DC pour alimenter les doubleurs de fréquences.

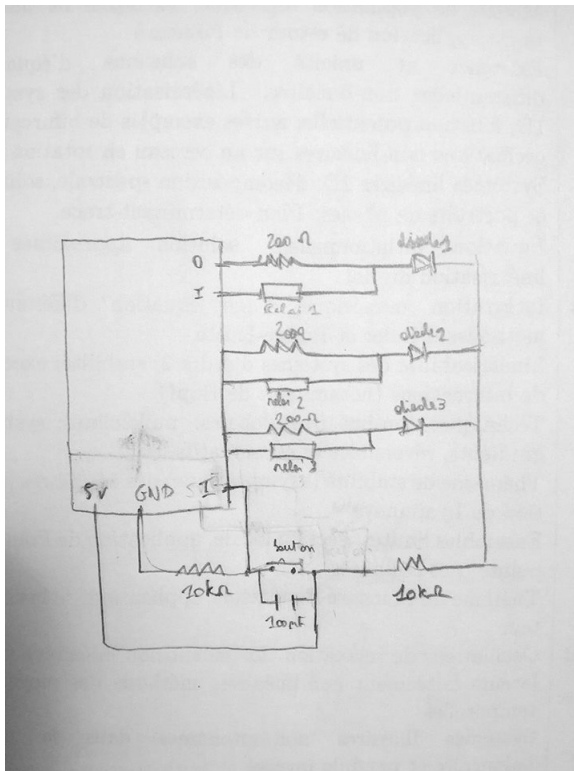


FIGURE 2 – Schéma en bloque zoom sur la partie connexion avec l'Arduino

Dans ce système, on a 3 relais :

- ° Relai 1 : Pour vérifier l'état du convertisseur AC/DC (s'il est fonctionnel)
- ° Relai 2 : Control de la sortie A de la diode(permet de vérifier si la diode laisse bien passer le courant venant de l'AC/DC)
- ° Relai 3 : Control de la sortie B de la diode(permet de vérifier si la diode laisse bien passer le courant venant de la backup batterie)

Remarque : Le Maser est continuellement alimenté par les deux sources (cas optimal)

## Programme :

Avant de programmer la logique du système, il faut lister les différents cas de figure qui vont engendrer des erreurs lors d'une panne.

Cas1: <b>relai 1</b> activé				
<b>relai2/relai3</b>	activé/activé	activé/désactivé	désactivé/activé	désactivé/désactivé
erreur	pas d'erreur	relai 3 défaillant	<b>relai 2</b> défaillant	pas d'alim pour la Diode
Cas2: <b>relai 1</b> désactivé				
<b>relai2/relai3</b>	activé/activé	activé/désactivé	désactivé/activé	désactivé/désactivé
erreur	erreur sur le <b>relai 1</b> et 2	<b>relai 3</b> défaillant+pas d'alim Diode	pas d'Alim venant de l' AC/DC	pas d'alim pour la Diode

FIGURE 3 – tableau des différentes erreurs possibles

- **Relai 1**
- **Relai 2**
- **Relai 3**

La logique du code est alors la suivante :

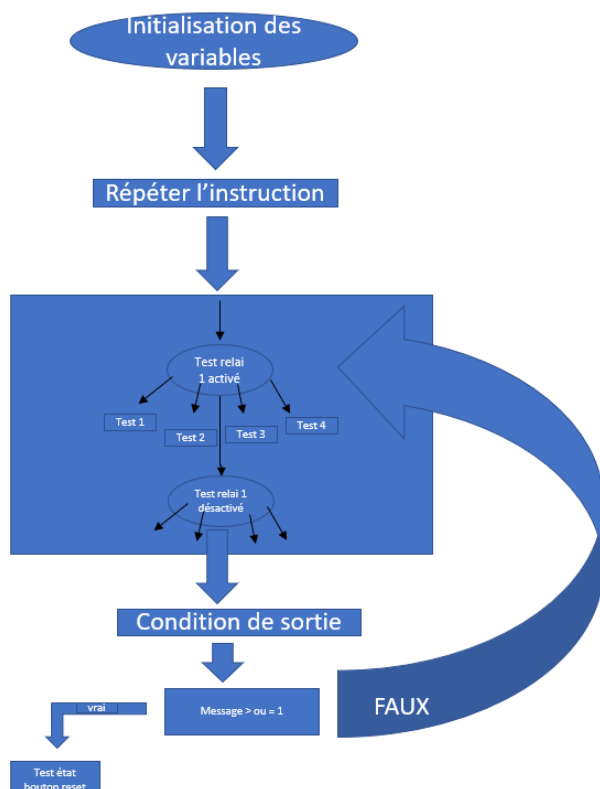


FIGURE 4 – Schéma de la logique du code

Sachant que pour effectuer un test sur les relai, on envoie un signal avec l'Arduino et on vérifie si l'état du relai est haut (alors il est activé) ou bas (désactivé). Après avoir repéré les erreurs, on envoie le mail et le sms via la fonction préalablement implémenté. On finit le code avec le test du bouton reset qui permet aux responsables de relancer le système lorsque la panne est résolu et d'envoyer une notification qui confirme que le problème est résolu.

Remarque : Afin d'éviter les spams, on choisit de passer par une variable qui limite l'envoi des messages d'erreurs. On a aussi une variable qui référence les différents types d'erreurs afin d'avoir une précision de l'erreur lors de l'envoi du mail/sms.