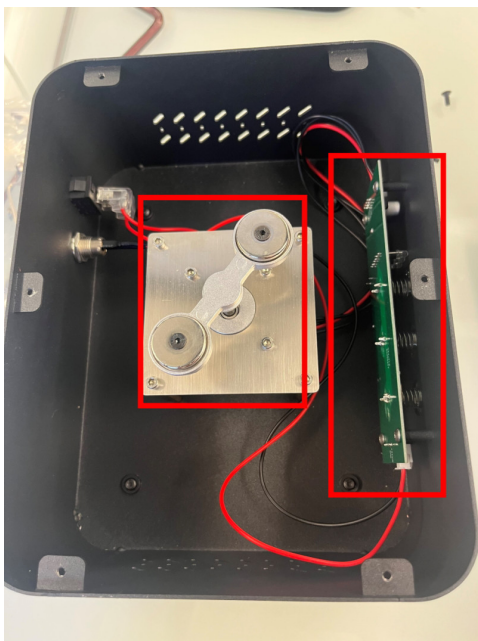


Sovol 3D (Lavage de résin)

"Upgrade"

Dans ce projet, on cherche à trouver une solution à la machine de lavage Sovol 3D qui s'arrête après quelques essais. Pour cela, on envisage à changer la mécanique de bouton (qui est actuellement faite avec une bouton de commutation) et la remplacer avec une système de bouton poussoir.

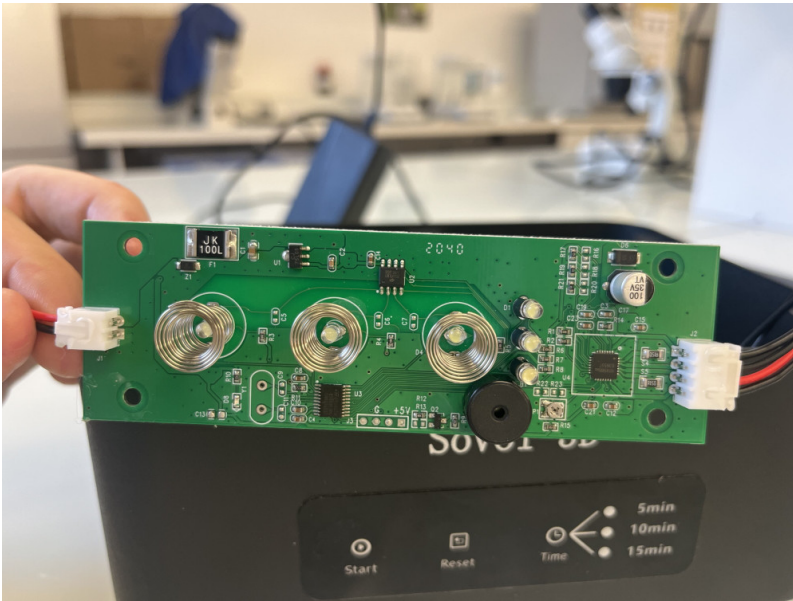
- Tension : 110V/220V 50/60HZ
- Tension de sortie : 24 V
- Courant du produit : 0,3 A
- Puissance maximale : 7,2 W



Il y a deux composants principaux de la machine qui semblent défectueux :

- Le moteur, au centre, qui est activé via le "bouton" de démarrage fixé à droite de la machine.
 - Après de multiples essais, il semble que le mécanisme moteur fonctionne bien, même s'il se met parfois à tourner sans même appuyer sur le bouton de démarrage...
- Soit, le deuxième composant est la carte fixée sous les "boutons" externes,
 - Cela semble être le composant qui cause réellement le problème, car il semble y avoir un mécanisme non naturel de boutons "sans contact" qui activent certaines fonctionnalités de la machine, à savoir les boutons de démarrage, de réinitialisation

et de minuterie. Le système de bouton en général semble être composé d'un ressort qui détecte le mouvement (destiné à un doigt mais semble que tout fonctionne vraiment) pour activer ladite fonction.



Le projet peut alors s'articuler autour de la recherche d'un moyen de tester les relais entre les boutons et le moteur, et peut-être d'un moyen de remplacer les boutons de type ressort par de véritables boutons cliquables.

Après quelques tests, on a confirmé que la machine s'arrête après même pas une minute (pour un réglage de 5 min) ...

On utilise un Arduino Nano 328 comme base de notre projet, et pour le codage on se base sur des recherches conduites sur Internet et les forums de Github.

Première ébauche (22.3.2023)

```
const int startBouton = 2; // un bouton sur la broche 2
const int resetBouton = 3; // un bouton sur la broche 3
const int timerBouton = 4; // un bouton sur la broche 4
const int moteurPin = 5; // un moteur sur la broche 5
```

```
void setup()
{
    // on met le bouton en entrée
    pinMode(startBouton, INPUT_PULLUP);
    pinMode(resetBouton, INPUT_PULLUP);
    pinMode(timerBouton, INPUT_PULLUP);
    pinMode(moteurPin, OUTPUT);
}
```

```

}

int start_etat;
int reset_etat;
int timer_etat;
int t = 1;

void loop()
{
    start_etat = digitalRead(startBouton);
    reset_etat = digitalRead(resetBouton);
    timer_etat = digitalRead(timerBouton);

    if(start_etat == HIGH)
        actionAppuiStart(); // le bouton est appuyé

    if(reset_etat == HIGH)
        break; // peut-être pas la meilleure solution : ajouter une fonction interrupt qui peut se faire même pendant
le delay

    if(timer_etat == HIGH)
        actionAppuiTimer(); // le bouton est appuyé

    delay(10);
}

void actionAppuiTimer(){

    if (t = 3)
        t = 1;
    else
        t += 1;

}

void actionAppuiStart(){

    if (t = 1)
        digitalWrite(moteurPin,HIGH); // le moteur tourne pendant 5 minutes
        delay(300000);

```

```
digitalWrite(moteurPin,LOW);

if (t = 2)
digitalWrite(moteurPin,HIGH); // le moteur tourne pendant 10 minutes
delay(600000);
digitalWrite(moteurPin,LOW);

else
digitalWrite(moteurPin,HIGH); // le moteur tourne pendant 15 minutes
delay(900000);
digitalWrite(moteurPin,LOW);

}
```

Réflexions

Ce code est relativement simple : on est allé chercher une algorithme de « comptage » sur Youtube. Nous nous sommes croisés avec un tutoriel d'Arduino qui permet de réaliser une programme qui effectue le comptage requise pour la partie « Timer » de la machine, car nous avons l'idée que c'est la fonction celui-ci qui pose problème pendant l'exécution de la session voulue (5,10,15 min).

Après avoir appris comment faire le code sur Arduino IDE, notre prochain étape était de trouver quels « ports » à utiliser sur la carte Arduino, car apparemment notre système requiert une module qui se connecte à la première sortie à droite de la carte. Dans le tutoriel, toutes les connections nécessaires sont introduites, et nous nous sommes débrouillés pour déterminer lesquels à utiliser dans notre cas.

Dans le code, on a aussi traité le sujet de relier ce système de « Timer » avec les boutons poussoirs qu'on veut introduire a la machine. Comme précédemment évoqué, les boutons actuelles de la machine peuvent influencer le démarrage de la machine, car son approche d'un système « à distance » crée une problème de « mal connexion » quand on veut commencer la procédure. Pour ceci, on a introduit une partie dans le code où, selon le bouton cliqué sur la machine, le « Timer » effectue des durées selon notre souhait.

Le code nécessite bien sûr des améliorations, il n'est pas du tout complet et il y a la possibilité qu'il soit moins complexe. Comme nous sommes toujours à L1, nos connaissances sont assez limitées, donc les autres réflexions sont à considérer.

Liens

<https://youtu.be/hriSLWYOMyM>

<https://youtu.be/EGNBmjVzZBE>

<https://youtu.be/gd1DUXZ8H34>

Revision #1

Created 19 May 2024 11:22:57 by Ouerfili Chaima

Updated 19 May 2024 11:22:57 by Ouerfili Chaima