



Compte rendu Séance 3

SYSTÈME D'ARROSAGE AUTOMATIQUE GÉRÉ PAR ESP32

Introduction

Le projet "Arrosage Automatique contrôlé par ESP32" est un projet tutoré par Christian Peter dans le cadre du projet d'électronique S7 ELSE 4 FISA. Ce projet est mené en collaboration par Neil Amrane, Salma Elfiache, Wissal Bellahcen et Khady Ndyé Diop.

Cet enseignement a pour but de développer un esprit de collaboration et une dynamique créative tout en favorisant l'autonomie dans la création d'un dispositif fonctionnel. Ce projet permet, en outre, de se familiariser avec l'électronique haut niveau et l'interconnexion de différents sous-systèmes électroniques en observant un projet dans sa globalité avec ses tenants et aboutissants.

L'objectif est de créer un système d'arrosage automatique permettant de contrôler 4 voies individuellement en programmant la durée d'arrosage de chaque voie et la fréquence d'arrosage des 4 voies. Ce même système intègre différents capteurs qui permettent de mieux gérer l'irrigation des zones couvertes par les circuits d'arrosage et de préserver la biodiversité en tenant compte des données géologiques et atmosphériques réelles et locales

Compte-rendu

Faire CR Projet S7 : aide aux différents membres du groupe sur programmation et compréhension des procédés; test des électrovanne sur banc de test avec alim DC : Pb de connection pinces crocos...; réflexion autour du schéma électronique avec prof ... Assez d'IO ? ; bilan de composants; début programmation écran LED.

1- Débriefing avec l'équipe des différentes avancées.

A ce stade du projet, les idées générales ont été posées et le cahier des charges a été validé par les membres du groupe. Pour l'instant quelques bouts de codes ont vu le jour mais aucun rendu concret.

Une partie de ma séance a été dédiée à aider les membres du groupe sur certaines erreurs de programmation, ainsi qu'à expliquer le processus et la logique de fonctionnement du système.

Cette étape est cruciale pour développer une dynamique de groupe efficace et ne pas perdre du temps sur la suite du projet.

2- Test des électrovannes

Nous avons poursuivi notre séance (avec Khadi) en testant les électrovannes sans pression afin de vérifier leur fonctionnement pratique.

Nous avons utilisé une alimentation DC paramétrée en 12V et nous avons alimenté l'électrovanne à l'aide de pince 'croco' et de câbles "Banane-Banane"

Nous avons rencontré des difficultés à alimenter l'électrovanne, du fait des connexions certainement usées par le temps. (recouper les câbles et installer des connecteurs sera probablement utile)

Le test a montré que l'électrovanne tirait ~120mA à l'ouverture et à la fermeture.

Cette valeur est inférieure à **la valeur théorique qui était de 500mA** à partir de la datasheet constructeur qui annonçait 12V DC 6W (soit 500mA de consommation)

On suppose que l'électrovanne atteint la consommation de 500 mA lorsque le circuit d'eau est sous pression.

Dans le cadre de ce projet, la contrainte de pression n'est pas incluse. En pratique, les différents niveaux (hauteurs des planchers à irriguer) imposant une pression suffisante pour acheminer l'eau jusqu'au niveau souhaité, nous obligerait à calculer (ou vérifier expérimentalement) la consommation en courant pour l'ouverture de l'électrovanne à la pression nécessaire au fonctionnement de l'installation prévue au cahier des charges.

3- Réflexion autour du schéma électronique

J'ai ensuite continué en travaillant avec notre professeur, M. Peter, sur le schéma électronique à transistors MOSFET. Nous avons choisi un montage à deux transistors à effet de champ, afin d'activer un transistor acceptant 12V 1A à partir d'un signal de 3,3V. Ce travail a été l'occasion de réactiver certaines notions autour des transistors.

Le premier MOSFET canal-N est activé par une tension VGS de 3,3V, le Drain de ce premier transistor est connecté à la grille du MOSFET canal-P qui est passant lorsque sa grille est à la masse.

4- Bilan GPIO

Le travail précédent a aussi mis en lumière le fait qu'avec cette approche, 8 GPIO étaient nécessaires pour actionner nos électrovannes. Nous avons donc, avec Wissal, réalisé le bilan des broches nécessaires à notre projet, afin de savoir si nous pouvions conserver ce montage ou si nous devons adopter une approche moins coûteuse en GPIO.

Ce bilan GPIO a montré que nous avons suffisamment d'IO sur le module ESP32 dont nous disposons pour poursuivre avec ce schéma.

Nous devons maintenant déterminer la référence des deux transistors à utiliser.

5- Début du codage de l'écran LED

J'ai entrepris de commencer la programmation de l'Interface Homme-Machine (IHM) afin de pouvoir intégrer les différentes informations au fur et à mesure.

```
#define SCREEN_LED 13

TFT_eSPI tft = TFT_eSPI();

void setup() {
  //Initialisation de l'écran
  tft.init();
  pinMode(SCREEN_LED, OUTPUT);
  digitalWrite(SCREEN_LED, HIGH);
  tft.setRotation(3); // portrait: 0 ou 2,

  // affichage d'un message à l'écran:
  tft.fillScreen(TFT_WHITE); // on efface t
  tft.setFreeFont(&FreeSansOblique12pt7b);
  tft.setCursor(20, 70); // position du déb
  tft.setTextSize(1);
  tft.setTextColor(TFT_BLACK, TFT_WHITE);
  tft.println("Notre équipe : ");
  delay(1000);
  tft.setTextColor(TFT_RED, TFT_WHITE);
  tft.println("Neil A");
  delay(1000);
  tft.println("Khady D");
  delay(1000);
  tft.println("Salma E");
  delay(1000);
  tft.println("Wissal B");
```

FIN DE COMPTE RENDU – RÉALISÉ LE 9/11/2025 – NEIL AMRANE