

Rapport Architecture

Boite aux lettres connectée

1°) Présentation du projet

L'objectif de ce projet est de réaliser une boîte aux lettres connectée qui au passage d'un facteur ou d'un livreur nous informe de la présence de courrier.

Pour ceci plusieurs actions devront être réaliser :

- La présence de courrier est correctement détectée
- Une LED s'allume en présence de courrier
- Le fanion se lève en présence de courrier
- L'afficheur LCD affiche « Passage » au minimum
- L'appuie sur reset permet immédiatement de revenir à l'état d'attente de courrier
- La LED de la carte clignote chaque seconde
- L'afficheur LCD affiche une heure courante et la met à jour chaque seconde
- L'afficheur LCD affiche l'heure courante après init depuis le PC
- L'afficheur LCD affiche l'heure de dernier passage du facteur
- La console du PC affiche la liste des dernières heures de passage du facteur

Pour réaliser ce projet nous avons utilisé le simulateur en ligne Tinkercad et aussi en se connectant en ssh à l'IUT.

Vous pouvez aussi trouver l'ensemble du travail sur github :

<https://github.com/cyril-lav/projet-archi>

2°) Choix réalisés

Pour remplir toutes contrainte nous avons décider réaliser certains choix parmi ceux proposés

Tout d'abord dans un premier temps voici les déclarations des entrées sorties de notre programme qui nous permettent de donner un nom à la cellule photo-électrique, la LED, le bouton, le servomoteur et l'écran lcd :

```
// declaration des pins des entrées/sorties
const int LED_PIN = 2;
const int SERVO_PIN = 9;
const int PHOTOCELL_PIN = A0;
const int BOUTON_PIN = 3;
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
Servo servoMoteur;
```

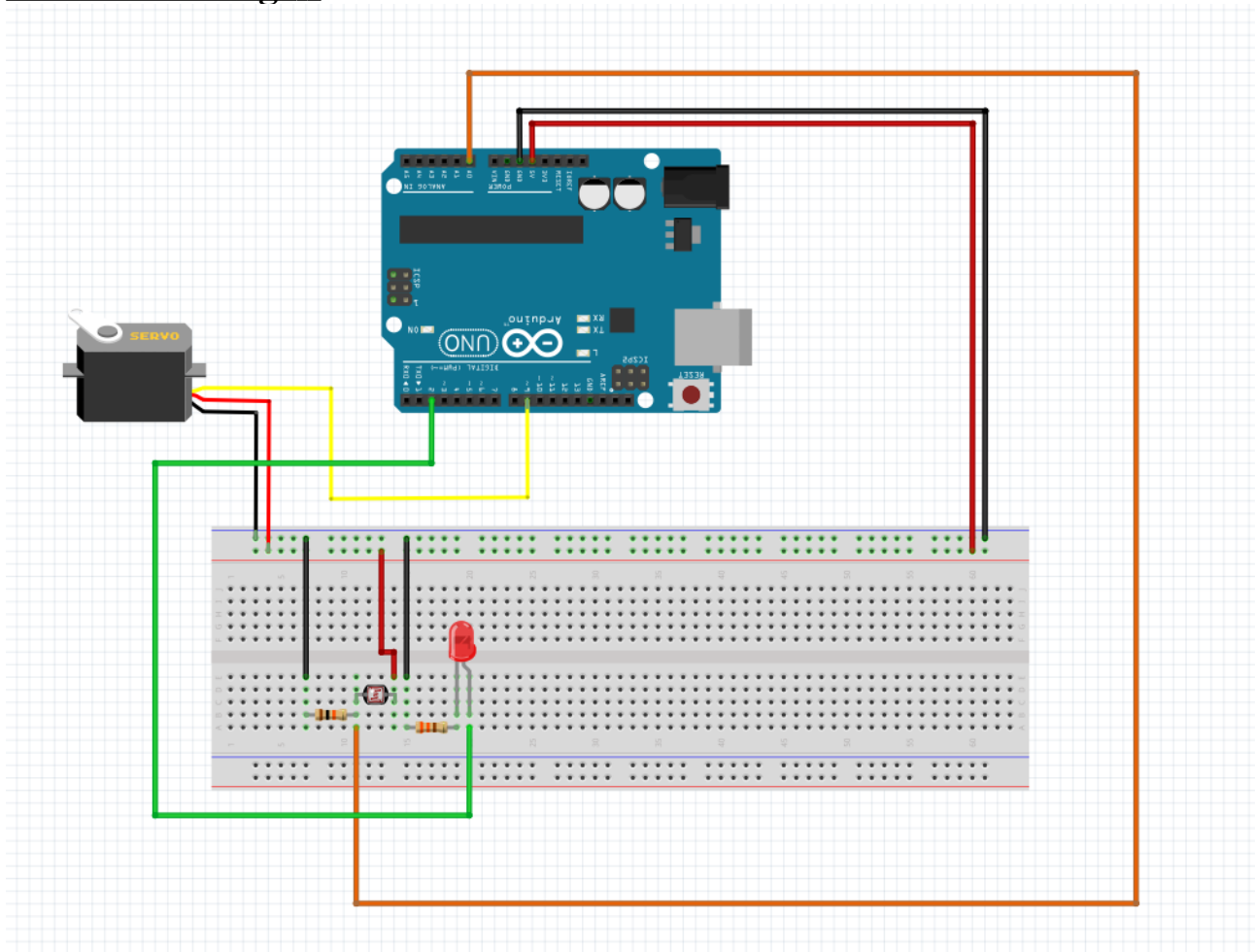
A) Détection et signalisation du courrier

Concernant le capteur nous avons eu le choix entre une cellule photo-électrique et un détecteur de pression.

Nous avons décidé de choisir une cellule photo-électrique en considérant que de la lumière passe car après de multiple test nous avons remarqué que le détecteur de pression moins précis lors du passage d'objets légers.

Avec cette cellule s'accompagne une LED qui s'allume lors de la détection d'un objet et un servomoteur servira de fanion et se lèvera à la détection.

Voici le montage :



Voici les fonctions (Testées en cours) :

```
void setup(){
  Serial.begin(9600);

  pinMode(LED_PIN,OUTPUT);
  pinMode(PHOTOCELL_PIN,INPUT);
  pinMode(13, OUTPUT);

  servoMoteur.attach(SERVO_PIN);

  attachInterrupt(BOUTON_PIN, reinitialiser, FALLING);

  initHorloge();
  initLCD();
  afficherHeureCourante();
}
```

Au démarrage du programme on définit nos entrées/sorties

```
void leverFanion(){
  servoMoteur.write(180); //baisse le fanion (servomoteur à 180°)
}

void allumerLed(){
  digitalWrite(LED_PIN, HIGH); //allume la led
}

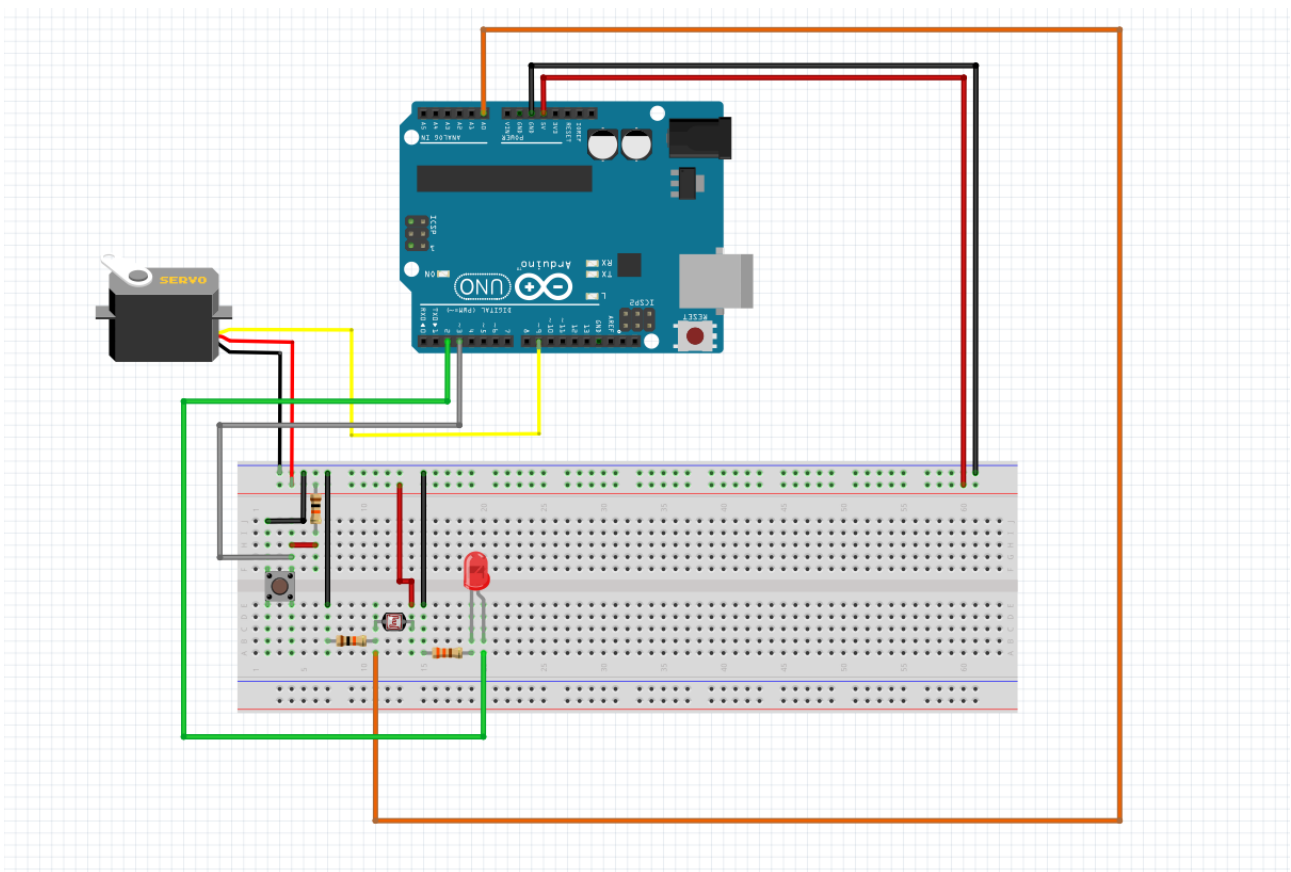
boolean verifierCourrier(){
  int valeurPhotocell = analogRead(PHOTOCELL_PIN);
  if(valeurPhotocell < 800){
    heuresPassage = heures;
    minutesPassage = minutes;
    afficherHeurePassage();
    leverFanion();
    allumerLed();
    return true;
  }
  return false;
}
```

Ici la cellule capte la lumière et renvoie une valeur qui quand elle est plus petite que 800 déclenche les 3 fonctions ci-dessus qui vont respectivement afficher l'heure du dernier passage (expliqué plus tard), lever le fanion et allumer la LED.

B) Réinitialisation du système

Pour faire revenir le système à son état d'origine nous avons décidé d'utiliser un bouton qui va fonctionner par interruption et non par scrutation car on veut que le reset du système soit rapide or la méthode par interruption permet cela car elle stop l'exécution du programme courant pour exécuter la fonction pour ensuite le reprendre.

Voici le montage :



Voici les fonctions :

```
void baisserFanion(){
    servoMoteur.write(0); //baisse le fanion (servomoteur à 0°)
}

void eteindreLed(){
    digitalWrite(LED_PIN, LOW); //éteint la led
}

void reinitialiser(){
    courrier = false;
    baisserFanion();
    eteindreLed();
}

void setup(){
    Serial.begin(9600);

    pinMode(LED_PIN,OUTPUT);
    pinMode(PHOTOCELL_PIN,INPUT);
    pinMode(13, OUTPUT);

    servoMoteur.attach(SERVO_PIN);

    attachInterrupt(BOUTON_PIN, reinitialiser, FALLING);

    initHorloge();
    initLCD();
    afficherHeureCourante();
}
```

A chaque pression du bouton une routine d'interruption se lance et exécute la fonction réinitialiser pour baisser le fanion et éteindre la LED.

C) Clignotement de la LED interne

Pour réaliser le clignotement de la LED interne à la carte nous avons utilisé la fonction millis d'arduino qui nous permet grâce à des calculs de la faire clignoter contrairement à un delay qui lui met en pause tout le programme.

Variables globales :

```
unsigned int tempsOn = 50;
unsigned int tempsOff = 950;
unsigned long tempsMemo = 0;
boolean commut = true;
```

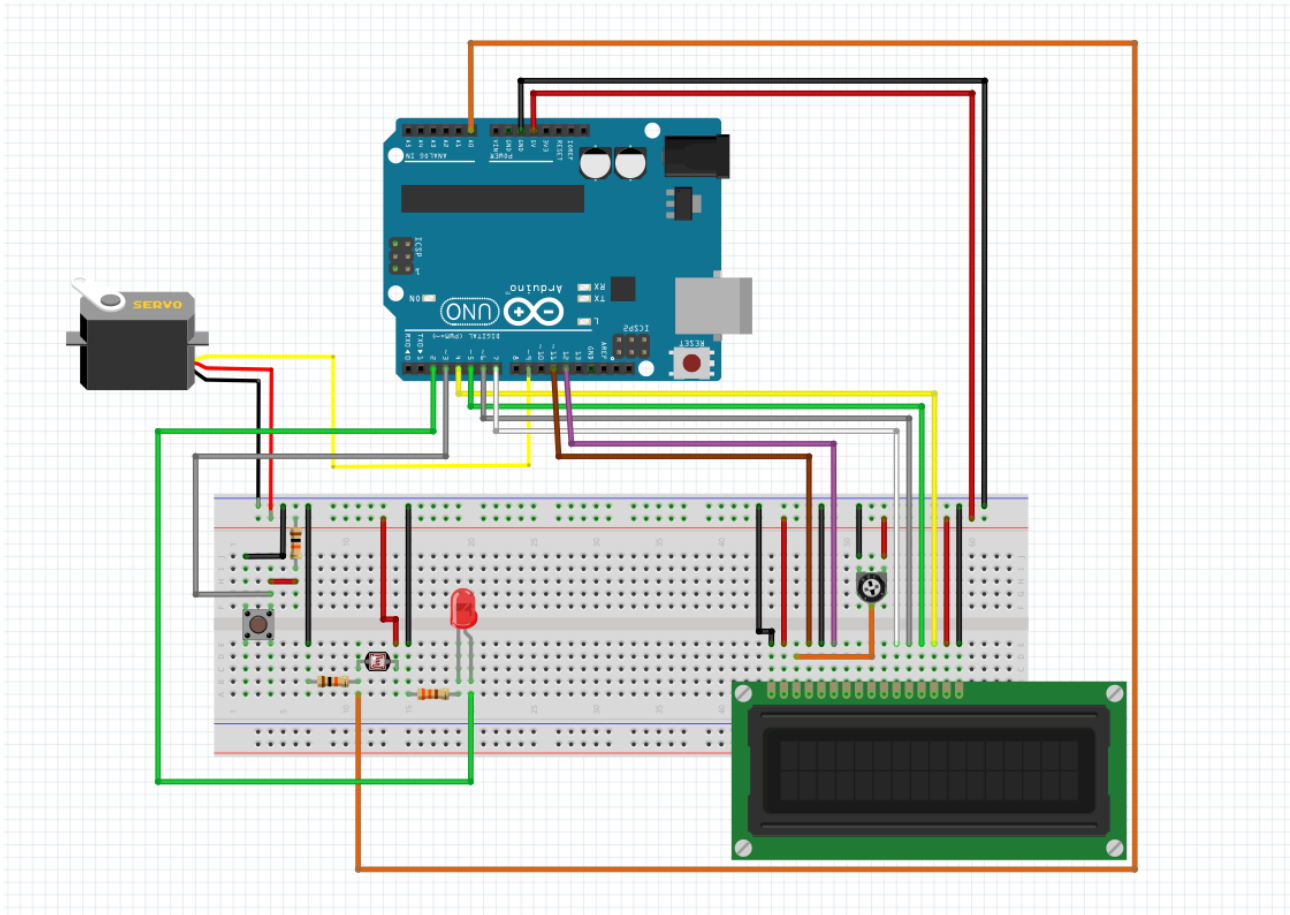
Voici la fonction (Testée en cours) :

```
void clignoterLed(){
    unsigned long temps = millis();
    if(temps >= tempsMemo){
        digitalWrite(13, commut); //inverse l'etat de la led
        if(commut)
            tempsMemo += tempsOn;
        else
            tempsMemo += tempsOff;
        commut = !commut;
    }
}
```

Grâce à la fonction millis qui retourne le temps écoulé en millisecondes depuis le démarrage de la carte, à chaque seconde écoulée la LED s'allume pendant 1/20 de seconde.

D) Affichage de l'heure

Voici le montage pour l'écran lcd :



Voici l'initialisation de l'écran :

```
void initLCD() {  
  lcd.clear();  
  lcd.begin(16,2);  
}
```


Voici les variables globales :

```
unsigned long tempsMs1;  
unsigned long tempsMs2 = 0;  
int heures;  
int minutes;  
int heuresPassage;  
int minutesPassage;
```

Voici les fonctions pour l'initialisation et l'affichage de l'heure de la carte :

```
void initHorloge(){  
    Serial.println("Saisir heures (hh) : ");  
    heures = saisieNombre();  
    Serial.println("Entrer minutes (mm) : ");  
    minutes = saisieNombre();  
    Serial.println(String(heures) + ":" + String(minutes));  
}  
  
int saisieNombre() {  
    String chaine = "";  
    while(true) {  
        while (Serial.available()) {  
            delay(3);  
            if (Serial.available() >0) {  
                char c = Serial.read();  
                chaine += c;  
            }  
        }  
  
        if(chaine != "") {  
            return chaine.toInt();  
        }  
    }  
}
```

Ici la fonction saisieNombre va lire caractère par caractère l'heure saisie par l'utilisateur.

```
void afficherHeureCourante(){  
    lcd.setCursor(0,0);  
    if(heures < 10)  
        lcd.print("0");  
    lcd.print(heures);  
    lcd.print("h");  
    if(minutes < 10)  
        lcd.print("0");  
    lcd.print(minutes);  
}
```

Cette fonction affiche l'heure qui est contenu dans les variables globales heures et minutes qui sont calculées ci-dessous

```
void calculHeure(){  
    tempsMs1 = millis();  
    if((tempsMs1 - tempsMs2) >= 60000){  
        tempsMs2 = tempsMs1;  
        minutes ++;  
    }  
    if(minutes >= 60){  
        heures ++;  
        minutes -= 60;  
        afficherHeureCourante(); //actualise l'affichage de l'heure sur l'écran lcd chaque minute  
    }  
    if(heures >= 24)  
        heures -= 24;  
}
```

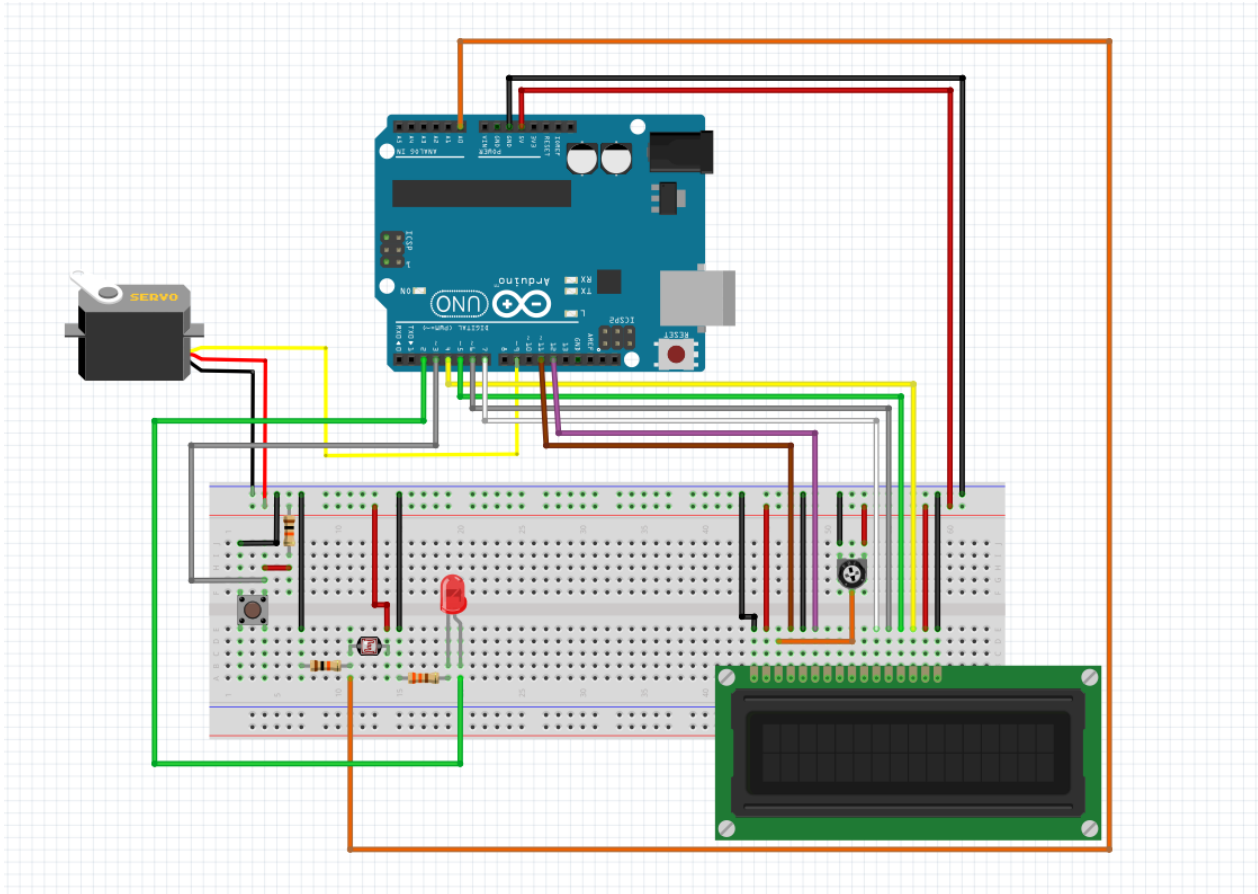
Cette fonction calcule l'heure grâce à la fonction millis qui décompte le temps depuis que la carte est allumé.

```
void afficherHeurePassage(){
    lcd.setCursor(0,1);
    lcd.print("Dernier passage : ");
    Serial.print("Passage : ");
    if(heures < 10){
        lcd.print("0");
        Serial.print("0");
    }
    lcd.print(heuresPassage);
    Serial.print(heuresPassage);
    lcd.print("h");
    Serial.print("h");
    if(minutes < 10){
        lcd.print("0");
        Serial.print("0");
    }
    lcd.print(minutesPassage);
    Serial.println(minutesPassage);
}
```

Cette fonction affiche l'heure du dernier passage sur le lcd puis sur le terminal du PC.

3°) Fonctionnement général

Voici le schéma final du projet :



Voici le setup ainsi que la loop :

```
void setup(){
    Serial.begin(9600);

    pinMode(LED_PIN,OUTPUT);
    pinMode(PHOTOCELL_PIN,INPUT);
    pinMode(13, OUTPUT);

    servoMoteur.attach(SERVO_PIN);

    attachInterrupt(BOUTON_PIN, reinitialiser, FALLING);

    initHorloge();
    initLCD();
    afficherHeureCourante();
}

void loop(){
    clignoterLed();
    calculHeure();
    if(!courrier)
        courrier = verifierCourrier();
}
```