

Machine à États Finis

➤ Choix des différents états

1. Qualité_Air_Bonne : Signal du capteur de qualité d'air
2. Température_Bonne : Signal du capteur de température
3. Commande_Manuelle : Interrupteur/application
4. Fenêtre_Ouverte : Fin de course du vérin en position ouverte
5. Fenêtre_Fermée : Fin de course du vérin en position fermée
6. Defaut_Capteur : Détection de panne
7. Surcharge_Moteur : Nouveau - Détection de courant excessif (blocage du vérin)

Sorties (adaptées pour le vérin) :

1. Alim_Vérit_ON : Activation de l'alimentation du vérin (relais ou MOSFET)
2. Sens_Extension : Commande pour l'extension du vérin (OUVERTURE)
3. Sens_Rétraction : Commande pour la rétraction du vérin (FERMETURE)
4. LED_Status : Indication visuelle
5. Alert : Notification utilisateur

➤ Implémentation possible avec arduino pour la vérification

```
// smart_window.ino - Version de prototypage

const int PIN_VERIN_EXTEND = 8;
const int PIN_VERIN_RETRACT = 9;
const int PIN_FIN.Course_OUVERT = 10;
```

```

const int PIN_FIN_COURSE_FERME = 11;
const int PIN_CAPTEUR_AIR = A0;

// Seuils à calibrer
const int SEUIL_AIR_MAUVAIS = 500;
const int TEMPS_MAX_MOUVEMENT = 30000; // 30 secondes

void setup() {
    Serial.begin(9600);
    pinMode(PIN_VERIN_EXTEND, OUTPUT);
    pinMode(PIN_VERIN_RETRACT, OUTPUT);
    pinMode(PIN_FIN_COURSE_OUVERT, INPUT_PULLUP);
    pinMode(PIN_FIN_COURSE_FERME, INPUT_PULLUP);

    Serial.println("Smart Window - Initialisation");
}

void loop() {
    machineEtats();
    delay(100);
}

void machineEtats() {
    static int etat = 0;
    static unsigned long chronoMouvement = 0;

    int airValue = analogRead(PIN_CAPTEUR_AIR);
    bool airMauvais = (airValue > SEUIL_AIR_MAUVAIS);
    bool finOuvert = digitalRead(PIN_FIN_COURSE_OUVERT);
    bool finFerme = digitalRead(PIN_FIN_COURSE_FERME);
}

```

```

switch(etat) {

    case 0:// INIT
        Serial.println("Etat: INIT");
        etat = 1;
        break;

    case 1:// ATTENTE_FERMEE
        if(!finFerme) {
            etat = 0; // Retour init si position incertaine
        } else if(airMauvais) {
            Serial.println("Air mauvais - Ouverture");
            verin_ouvrir();
            chronoMouvement = millis();
            etat = 2;
        }
        break;

    case 2:// OUVERTURE_EN_COURS
        if(finOuvert) {
            Serial.println("Fenetre ouverte");
            verin_stop();
            etat = 3;
        } else if(millis() - chronoMouvement > TEMPS_MAX_MOUVEMENT) {
            Serial.println("ERREUR: Timeout ouverture");
            verin_stop();
            etat = 99; // Etat erreur
        }
        break;
}

```

```

case 3: // ATTENTE_OUVERTE
if(!airMauvais) {
    Serial.println("Air bon - Fermeture");
    verin_fermer();
    chronoMouvement = millis();
    etat = 4;
}
break;

case 99: // ERREUR
verin_stop();
Serial.println("ETAT ERREUR - Intervention requise");
break;
}

void verin_ouvrir() {
    digitalWrite(PIN_VERIN_EXTEND, HIGH);
    digitalWrite(PIN_VERIN_RETRACT, LOW);
}

void verin_fermer() {
    digitalWrite(PIN_VERIN_EXTEND, LOW);
    digitalWrite(PIN_VERIN_RETRACT, HIGH);
}

void verin_stop() {
    digitalWrite(PIN_VERIN_EXTEND, LOW);
    digitalWrite(PIN_VERIN_RETRACT, LOW); }

```

