```
/* Constantes pour les broches */
const byte TRIGGER_PIN = 2; // Broche TRIGGER
const byte ECHO_PIN = 3;    // Broche ECHO

/*servo*/
#include <Servo.h>
Servo myServo;

/* Constantes pour le timeout */
const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m Ã  340m/s

/* Vitesse du son dans l'air en mm/us */
const float SOUND_SPEED = 340.0 / 1000;

const byte eVanne1opened = 7; //led verte vanne 1
const byte eVanne2opened = 8; //led verte vanne 2
const byte eVanne1closed = 12; //led rouge vanne 1
const byte eVanne2closed = 13; //led rouge vanne 2

const byte bp1 = 10;
const byte switchC= 9;


bool initState= true;
bool liquid1State= false;
bool liquid2State= false;

bool isGlass0= true;   //boolÃ©en indiquant la prÃ©sence d'un verre en
position 0
bool isGlass1= false;   //boolÃ©en indiquant la prÃ©sence d'un verre en
position 1
bool isGlass2= false;   //boolÃ©en indiquant la prÃ©sence d'un verre en
position 2

bool switch1= false;    //boolÃ©en indiquant la position de
l'interrupteur
bool dcy= false;        //boolÃ©en indiquant l'appui ou non sur le BP
bool old=false;         //boolÃ©en permettant de dÃ©terminer si le
versage du liquide a commencÃ©

double t1i,t2i;         //compteurs de versage
double dT;
double currentTime;     //temps actuel

void setup() {
  Serial.begin(115200);

  pinMode(eVanne1opened,OUTPUT);
  pinMode(eVanne2opened,OUTPUT);
  pinMode(eVanne1closed,OUTPUT);
  pinMode(eVanne2closed,OUTPUT);
```

```
  pinMode(bp1,INPUT);
  pinMode(switchC,INPUT);

  pinMode(TRIGGER_PIN, OUTPUT);
  digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit Ãªtre
Ã  LOW au repos
  pinMode(ECHO_PIN, INPUT);

  digitalWrite(eVanne1opened, LOW);
  digitalWrite(eVanne1closed, HIGH);
  digitalWrite(eVanne2opened, LOW);
  digitalWrite(eVanne2closed, HIGH);


  myServo.attach(4);
}

//Fonctions de rotations du servomoteur

void rotate0to1(){
  for (int i=80;i<=160;i++){
    myServo.write(i);
    delay(10);
  }
}
void rotate0to2(){
  for (int i=80;i>=0;i--){
    myServo.write(i);
    delay(10);
  }
}
void rotate1to0(){
  for (int i=160;i>=80;i--){
    myServo.write(i);
    delay(10);
  }
}
void rotate2to0(){
  for (int i=0;i<=80;i++){
    myServo.write(i);
    delay(10);
  }
}
////////////////////////////////////////////////

void loop() {

    /* 1. Lance une mesure de distance en envoyant une impulsion HIGH
de 10Âµs sur la broche TRIGGER */
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
```

```
  /* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et
son Ã©cho (si il existe) */
  long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);

  /* 3. Calcul la distance Ã  partir du temps mesurÃ© */
  float distance_mm = measure / 2.0 * SOUND_SPEED;

  if (distance_mm>50 && distance_mm<90){
    digitalWrite(eVanne1opened, HIGH); //allumer verte1
    digitalWrite(eVanne1closed, LOW); //eteindre orange1
    digitalWrite(eVanne2opened, LOW); //eteindre verte2
    digitalWrite(eVanne2closed, HIGH); //allumer orange2

  }

  if ((distance_mm<50)||(distance_mm>90 &&
distance_mm<250)||(distance_mm>290)){
    digitalWrite(eVanne1opened, LOW); //eteindre verte1
    digitalWrite(eVanne1closed, HIGH); //allumer orange1
    digitalWrite(eVanne2opened, LOW); //eteindre verte2
    digitalWrite(eVanne2closed, HIGH); //allumer orange2
  }

  if (distance_mm>250 && distance_mm<290){
    digitalWrite(eVanne1opened, LOW); //eteindre verte1
    digitalWrite(eVanne1closed, HIGH); //allumer orange1
    digitalWrite(eVanne2opened, HIGH); //allumer verte2
    digitalWrite(eVanne2closed, LOW); //eteindre orange2

  }

  currentTime = millis();
  switch1=digitalRead(switchC);
  dcy=digitalRead(bp1);

  if (initState){ //Ã©tat initial

    if (dcy && isGlass0){
      Serial.print("distance_mm");
      if (switch1){
        liquid1State=true;
        initState=false;
        rotate0to1();
      }
      else{
        liquid2State=true;
        initState=false;
        rotate0to2();
      }
    }
  }
```

```
  if (liquid1State){ //versement liquide 1

    if( isGlass1 && old==false){
      digitalWrite(eVanne1opened, HIGH);
      digitalWrite(eVanne1closed, LOW);
      t1i=currentTime;
    }
    dT=currentTime-t1i; }
    else if( (isGlass1 && dT>=2000) || isGlass1==false){
      digitalWrite(eVanne1opened, LOW);
      digitalWrite(eVanne1closed, HIGH);
      liquid1State=false;
      switch (switch1){
        case true: rotate1to0();
        rotate0to2();
        liquid2State=true;
        break;
        case false: rotate1to0();
        initState=true;
        break;
      }
    }

    if (liquid2State){//versement liquide 2

    if( isGlass1 && old==false){
      digitalWrite(eVanne2opened, HIGH);
      digitalWrite(eVanne2closed, LOW);
      t2i=currentTime;
    }
    dT=currentTime-t2i;
    if( (isGlass1 && dT>=2000) || isGlass1==false){
      digitalWrite(eVanne2opened, LOW);
      digitalWrite(eVanne2closed, HIGH);
      liquid2State=false;
      switch (switch1){
        case true: rotate2to0();
        rotate0to1();
        liquid1State=true;
        break;
        case false: rotate2to0();
        initState=true;
        break;
      }
    }
  }

}
```