

```

//define pins
#define water_level_max 5
#define water_level_min 6
#define relay_pump 7
#define relay_solenoid 8
#define lightSensor 10
#define relay_fan 9

int priming_time = 3000; // Priming time
const unsigned long interval_flood_drain = 21600000; // 6 hours-flood-drain
interval
unsigned long flood_drain_Millis = 0;

void setup() {
  pinMode(water_level_max, INPUT);
  pinMode(water_level_min, INPUT);
  pinMode(lightSensor, INPUT);
  pinMode(relay_pump, OUTPUT);
  pinMode(relay_fan, OUTPUT);
  pinMode(relay_solenoid, OUTPUT);

  digitalWrite(relay_pump, HIGH);
  digitalWrite(relay_solenoid, HIGH);
  digitalWrite(relay_fan, HIGH);
}

void loop()
{
  unsigned long currentMillis = millis();
  int lightSensorReading = digitalRead(lightSensor);
  if (currentMillis - flood_drain_Millis >= interval_flood_drain)
  {
    flood_drain_Millis = currentMillis;
    if (lightSensorReading == 0) //if light is on
    {
      flood_drain();
    }
    else
    {
      fanControl();
    }
  }
  else
  {
    fanControl();
  }
}

void flood_drain()
{
  unsigned long delay_drain = 50000; //50 seconds

  //PRIMING
  digitalWrite(relay_pump, LOW); //pump ON
  delay(priming_time);
  digitalWrite(relay_pump, HIGH); //pump OFF
}

```

```

//FLOODING
int Liquid_level_max = digitalRead(water_level_max);

while (Liquid_level_max == 0)
{
  Liquid_level_max = digitalRead(water_level_max);
}

//DRAINING
int Liquid_level_min = digitalRead(water_level_min);

while (Liquid_level_min == 1)
{
  digitalWrite(relay_pump, LOW); //pump ON
  Liquid_level_min = digitalRead(water_level_min);
}

delay(delay_drain);
digitalWrite(relay_pump, HIGH);
delay(1000);

//SIPHON BRAKER
digitalWrite(relay_solenoid, LOW);
delay(13000); //valve open for 13 seconds
digitalWrite(relay_solenoid, HIGH);

return;
}

void fanControl()
{
  int lightSensorReading = digitalRead(lightSensor);
  if (lightSensorReading == 0) //light is ON
  {
    digitalWrite(relay_fan, LOW); //switch ON fan
    return;
  }
  else
  {
    digitalWrite(relay_fan, HIGH); //switch OFF fan
    return;
  }
}
}

```