```
#include <Servo.h>
#include <avr/sleep.h>

const int soilMoisturePin = A0;
const int thresholdPin = A1;
const int pumpPin = 9;
const int buzzerPin = 7;
const int ledPin = 6;
const int buttonPin = 2;
Servo servoMotor;

volatile bool wakeUp = false;

void setup() {
  pinMode(soilMoisturePin, INPUT);
  pinMode(thresholdPin, INPUT);
  pinMode(pumpPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  servoMotor.attach(10);

  attachInterrupt(digitalPinToInterrupt(buttonPin), wakeUpNow, LOW);

  // Initial state
  digitalWrite(pumpPin, LOW);
  servoMotor.write(0);
}
```

```
void loop() {
  // Enter sleep mode if the button is not pressed
  if (!wakeUp) {
    enterSleep();
  }

  // Read soil moisture and threshold
  int soilMoisture = analogRead(soilMoisturePin);
  int threshold = analogRead(thresholdPin);

  // Water the plants if soil moisture is below the threshold
  if (soilMoisture < threshold) {
    digitalWrite(pumpPin, HIGH);
    servoMotor.write(90);
    provideFeedback(true);  // Watering starts
    delay(5000);            // Watering duration
    digitalWrite(pumpPin, LOW);
    servoMotor.write(0);
    provideFeedback(false); // Watering stops
  }

  // Wait 1 second before next check
  delay(1000);
  wakeUp = false;
}

void provideFeedback(bool watering) {
```

```
  // Short beep and LED blink for feedback
  for (int i = 0; i < 3; i++) {
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(ledPin, HIGH);
    delay(100);
    digitalWrite(buzzerPin, LOW);
    digitalWrite(ledPin, LOW);
    delay(100);
  }
}

void enterSleep() {
  set_sleep_mode(SLEEP_MODE_PWR_DOWN);
  sleep_enable();
  attachInterrupt(digitalPinToInterrupt(buttonPin), wakeUpNow, LOW);
  sleep_mode();

  // The program continues from here after waking up
  sleep_disable();
  detachInterrupt(digitalPinToInterrupt(buttonPin));
}

void wakeUpNow() {
  wakeUp = true;
}
```