Water Quality Monitoring with Arduino:

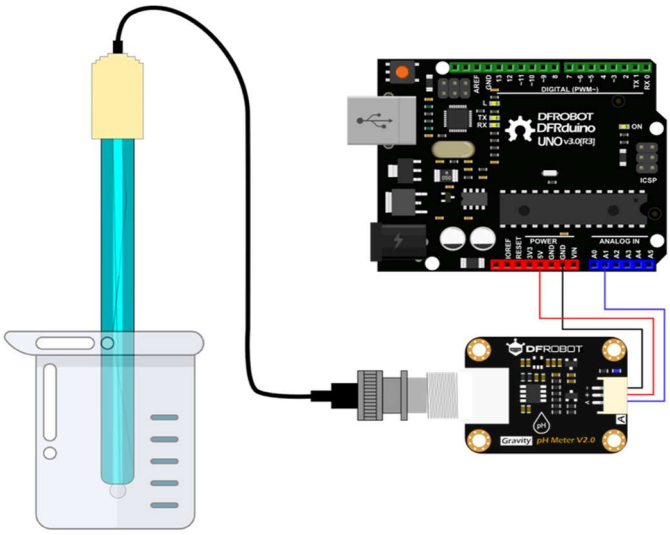
```
// Example code for Arduino pH sensor

int sensorPin = A0;  // Analog pin connected to the pH sensor

float pHValue;       // Variable to store pH value


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


void loop() {
  int sensorValue = analogRead(sensorPin);
  pHValue = /* Convert sensor value to pH */;
  Serial.print("pH value: ");
  Serial.println(pHValue);
  delay(1000);  // Update every second
}
```



Coffee Machine with Raspberry Pi:

```
import RPi.GPIO as GPIO
import time

# Set up GPIO using BCM numbering
GPIO.setmode(GPIO.BCM)
# Set the pin connected to the relay
coffee_relay_pin = 18
GPIO.setup(coffee_relay_pin, GPIO.OUT)

# Function to turn on the coffee machine
def turn_on_coffee_machine():
    GPIO.output(coffee_relay_pin, GPIO.HIGH)
    print("Coffee machine turned ON")

# Function to turn off the coffee machine
def turn_off_coffee_machine():
    GPIO.output(coffee_relay_pin, GPIO.LOW)
    print("Coffee machine turned OFF")

# Example usage
try:
    turn_on_coffee_machine()
    time.sleep(10)  # Let the coffee machine run for 10 seconds
    turn_off_coffee_machine()
except KeyboardInterrupt:
    GPIO.cleanup()
```

Current Sensor for Energy Consumption with Raspberry Pi

```
import RPi.GPIO as GPIO

# Set up GPIO using BCM numbering
GPIO.setmode(GPIO.BCM)
# Set the pin connected to the current sensor
current_sensor_pin = 17
GPIO.setup(current_sensor_pin, GPIO.IN)

# Function to read current sensor value
def read_current():
    # Read analog value from the pin (you might need ADC if it's an analog sensor)
    return GPIO.input(current_sensor_pin)

# Example usage
try:
    while True:
        current_value = read_current()
        print("Current value:", current_value)
except KeyboardInterrupt:
    GPIO.cleanup()
```

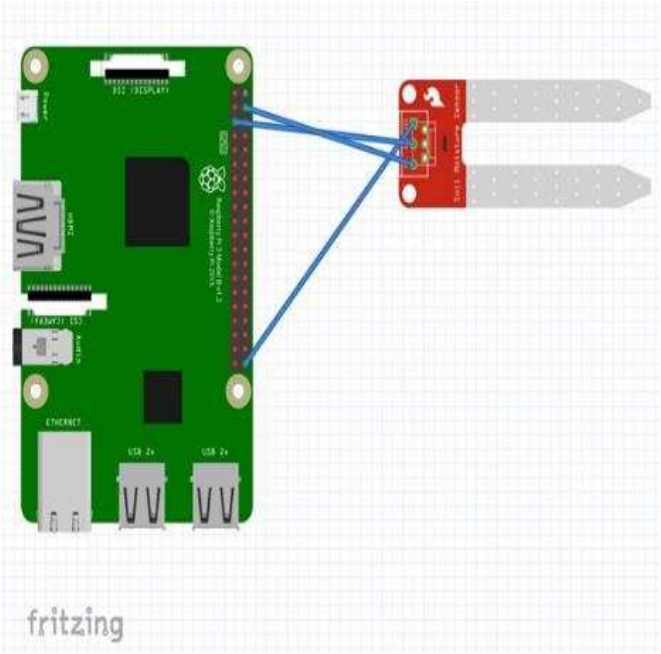
Soil Moisture Sensor with Raspberry Pi

```
import RPi.GPIO as GPIO
import time

# Set up GPIO using BCM numbering
GPIO.setmode(GPIO.BCM)
# Set the pin connected to the soil moisture sensor
moisture_sensor_pin = 17
GPIO.setup(moisture_sensor_pin, GPIO.IN)

# Function to read soil moisture level
def read_moisture():
    return GPIO.input(moisture_sensor_pin)

# Example usage
try:
    while True:
        moisture_value = read_moisture()
        if moisture_value == GPIO.LOW:
            print("Soil is dry")
        else:
            print("Soil is moist")
        time.sleep(1)  # Update every second
except KeyboardInterrupt:
    GPIO.cleanup()
```



Servo Motor Control with Raspberry Pi

```
import RPi.GPIO as GPIO
import time

# Set up GPIO using BCM numbering
GPIO.setmode(GPIO.BCM)
# Set the pin connected to the servo motor
servo_pin = 18
GPIO.setup(servo_pin, GPIO.OUT)

# Create PWM object
pwm = GPIO.PWM(servo_pin, 50)  # 50 Hz frequency for servo control

# Function to set servo angle
def set_angle(angle):
    duty_cycle = angle / 18.0 + 2.5  # Convert angle to duty cycle
    pwm.ChangeDutyCycle(duty_cycle)
    time.sleep(1)  # Give time for servo to reach the position

# Example usage
try:
    pwm.start(0)  # Start PWM with 0% duty cycle
    while True:
        set_angle(0)  # Rotate to 0 degrees
        time.sleep(2)
        set_angle(90)  # Rotate to 90 degrees
        time.sleep(2)
        set_angle(180)  # Rotate to 180 degrees
        time.sleep(2)
except KeyboardInterrupt:
    pwm.stop()
    GPIO.cleanup()
```

Servo Motor Control with Arduino

```
#include <Servo.h>

// Create a servo object
Servo myservo;

// Pin number to which the servo signal wire is connected
int servoPin = 9;

void setup() {
  // Attach the servo to the pin
  myservo.attach(servoPin);
}

void loop() {
  // Sweep the servo from 0 to 180 degrees
  for (int angle = 0; angle <= 180; angle++) {
    myservo.write(angle);
    delay(15);  // Adjust speed of the sweep
  }

  // Sweep the servo from 180 to 0 degrees
  for (int angle = 180; angle >= 0; angle--) {
    myservo.write(angle);
    delay(15);  // Adjust speed of the sweep
  }
}
```