

```

/*
TheRemoteCode.ino

This program uses the ArduinoBLE library to set-up an Arduino Nano 33 BLE Sense
as a central device and looks for a specified service and characteristic in a
peripheral device. If the specified service and characteristic is found in a
peripheral device, the last detected value of the joystick is written in the specified characteristic.

The circuit:
- Arduino Nano 33 BLE Sense.

THIS IS FOR THE REMOTE SIDE
This improved version controls the dropper post

*/

#include <ArduinoBLE.h>
#include <Arduino_APDS9960.h>
#include <ezButton.h>

const char* deviceServiceUuid = "19b10000-e8f2-537e-4f6c-d104768a1214";
const char* deviceServiceCharacteristicUuid = "19b10001-e8f2-537e-4f6c-d104768a1214";

int gesture = 0;
int oldGestureValue = 1;
int VRy = 0;
int SW = 0;
int buttonstate = 0;
const int SW_pin_in = 2;

void setup() {
  pinMode(SW_pin_in, INPUT_PULLUP);
  Serial.begin(9600);
  // while (!Serial); //uncomment this for use without

  if (!BLE.begin()) {
    Serial.println("Starting Bluetooth® Low Energy module failed!");
    while (1);
  }

  BLE.setLocalName("Nano 33 BLE (Central)");
  BLE.advertise();

  Serial.println("Arduino Nano 33 BLE Sense (Central Device)");
  Serial.println(" ");
}

void loop() {
  connectToPeripheral();
}

void connectToPeripheral() {
  BLEDevice peripheral;

  Serial.println("- Discovering peripheral device...");

  do
  {
    BLE.scanForUuid(deviceServiceUuid);
    peripheral = BLE.available();
  } while (!peripheral);

  if (peripheral) {
    Serial.println("Peripheral device found!");
    Serial.print("Device MAC address: ");
    Serial.println(peripheral.address());
    Serial.print("Device name: ");
    Serial.println(peripheral.localName());
    Serial.print("Advertised service UUID: ");
    Serial.println(peripheral.advertisedServiceUuid());
    Serial.println(" ");
    BLE.stopScan();
    controlPeripheral(peripheral);
  }
}

void controlPeripheral(BLEDevice peripheral) {
  Serial.println("- Connecting to peripheral device...");

  if (peripheral.connect()) {
    Serial.println("Connected to peripheral device!");
    Serial.println(" ");
  } else {
    Serial.println("Connection to peripheral device failed!");
    Serial.println(" ");
    return;
  }

  Serial.println("- Discovering peripheral device attributes...");
  if (peripheral.discoverAttributes()) {
    Serial.println("Peripheral device attributes discovered!");
    Serial.println(" ");
  } else {
    Serial.println("Peripheral device attributes discovery failed!");
    Serial.println(" ");
    peripheral.disconnect();
    return;
  }

  BLECharacteristic gestureCharacteristic = peripheral.characteristic(deviceServiceCharacteristicUuid);

  if (!gestureCharacteristic) {
    Serial.println("Peripheral device does not have gesture_type characteristic!");
    peripheral.disconnect();
    return;
  } else if (!gestureCharacteristic.canWrite()) {
    Serial.println("Peripheral does not have a writable gesture_type characteristic!");
    peripheral.disconnect();
    return;
  }

  while (peripheral.connected()) { // this is the improitant stuff
    gesture = gestureDetection();
    if (oldGestureValue != gesture) {
      Serial.print("Writing value to gesture_type characteristic: ");
      Serial.println(gesture);
      gestureCharacteristic.writeValue((byte)gesture);
      Serial.println("Writing value to gesture_type characteristic done!");
      Serial.println(" ");
      oldGestureValue = gesture; //now here
    }
  }
  Serial.println("- Peripheral device disconnected!");
}

int gestureDetection() {
  buttonstate = digitalRead(SW_pin_in);
  delay(10);
  VRy = analogRead(A1);
  gesture = map(VRy, 0, 1000, 0, 2); //changed from 1023 bec needed to deal with 3.3 v not hitting 1023 consisantly
  delay(10);
  if (buttonstate == 0) {
    delay(10); // if switch is tripped, then write gesture as 3
    gesture = 3;
    delay(10);
  }
  delay(10);
  return gesture;
}
```