Microcontroller Experiments using Arduino or MSP430:

a. Touch sensor

b. Tracking sensor

c. Tap sensor

a. Touch sensor

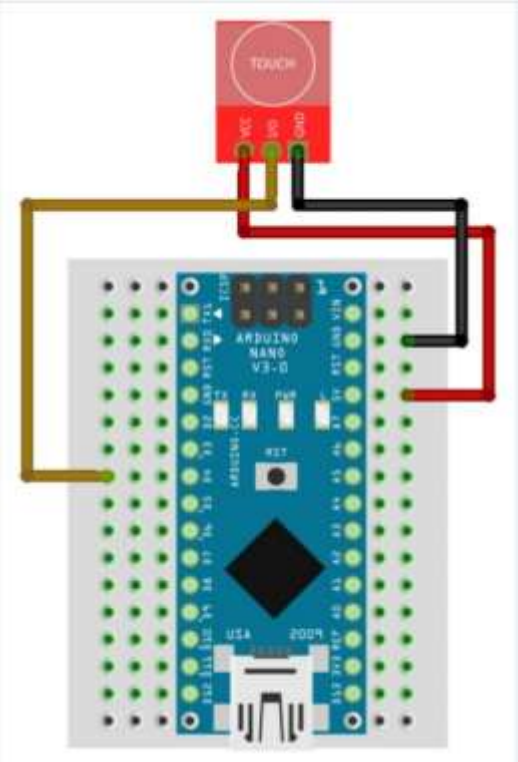
Components Required:

TTP223B

Arduino

Jumper wires

Bread Board



Program:

```
#define BUTTON_PIN 4
struct touch {
        byte wasPressed = LOW;
        byte isPressed = LOW;
};
touch touch;
void setup()
{
        pinMode(BUTTON_PIN, INPUT);
        Serial.begin(115200);
}
void loop()
{
        touch.isPressed = isTouchPressed(BUTTON_PIN);
        if (touch.wasPressed != touch.isPressed) {
          Serial.println("Touch pressed");
        }
        touch.wasPressed = touch.isPressed;
}
bool isTouchPressed(int pin)
```

```
{
        return digitalRead(pin) == HIGH;
}
```

b. Tracking sensor

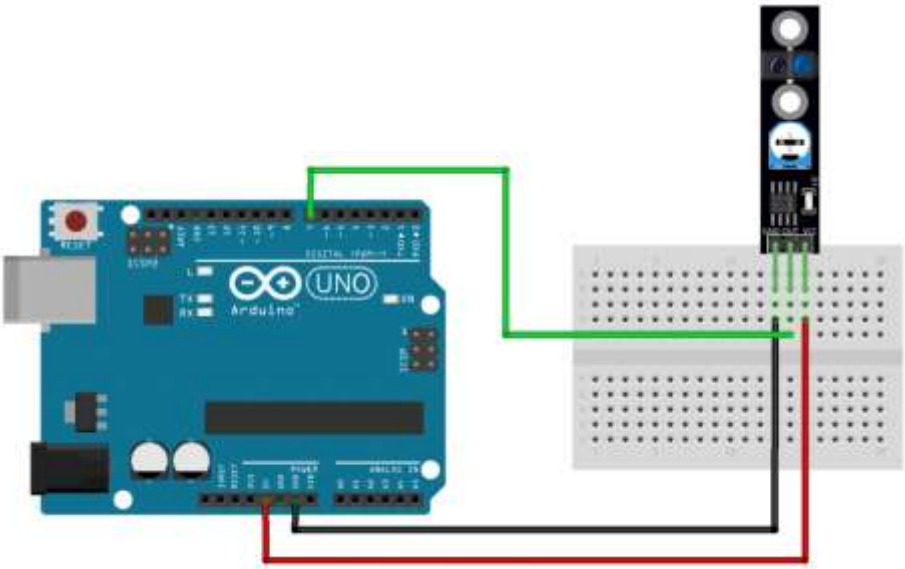
Components Required:

KY-033 Line Tracking Sensor Arduino circuit and Programming

Arduino

Jumper wires

Bread Board

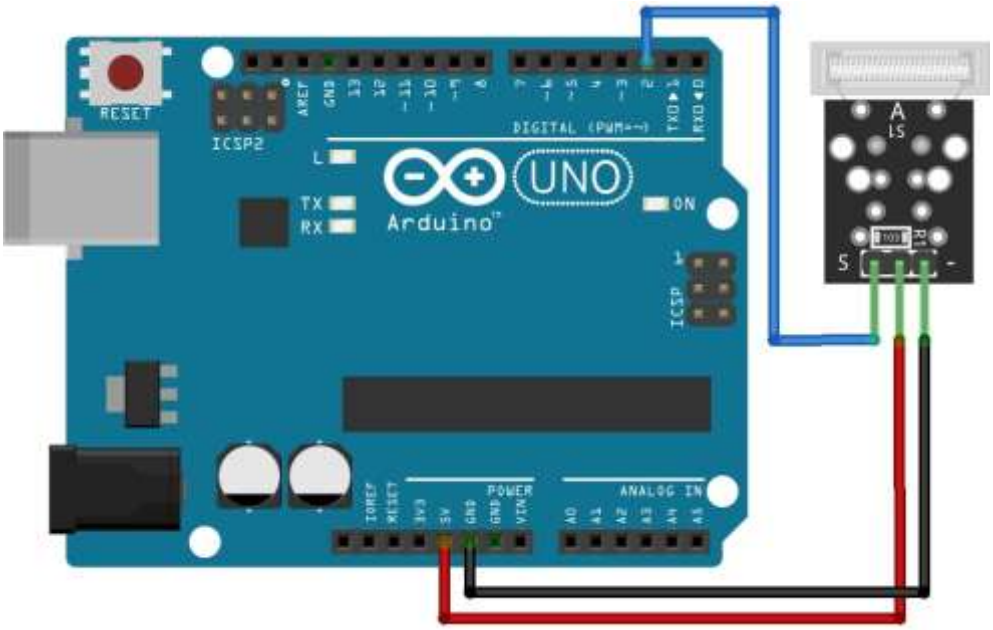


Program

```
int sensorPin = 7;  // line detection sensor interface
int val;            // variable to store sensor reading

void setup() {
  pinMode(sensorPin,INPUT);  // define sensor as input
  Serial.begin(9600);     // initialize serial communication with PC
}

void loop() {
  val = digitalRead(sensorPin); // read value from sensor

  if (val == HIGH) {
    Serial.println("Line detected");
  } else {
    Serial.println("Line NOT detected");
  }

  delay(500);
}
```

c. Tap Sensor
Components Required:

- Arduino board (e.g., Arduino Uno)
- Tap Module
- Jumper wires
- Breadboard (optional)



Program:

```
#define tapSensorPin 2 // Change this to the pin you're using
volatile bool tapped = false; // Flag to indicate tap detection

void setup()
{
  pinMode(tapSensorPin, INPUT_PULLUP); // Set the sensor pin as input with internal pull-up resistor
  attachInterrupt(digitalPinToInterrupt(tapSensorPin), tapDetected, RISING); // Attach interrupt to the pin for rising edge detection
  Serial.begin(9600); // Initialize serial communication
}
void loop() {
  if (tapped) {
    Serial.println("Tap detected!");
    tapped = false; // Reset the tap flag
    // Add actions or responses when a tap is detected
  }
  delay(100); // Delay for stability and to prevent rapid consecutive detections
}
void tapDetected() {
  tapped = true; // Set the tap flag when a rising edge is detected
}
```