

## Projekt Nr. 33: Bereiche auf dem Touchscreen ansprechen

---

Zwischen der Kunststoffabdeckung und dem Glasboden des Touchscreens befinden sich zwei elektrisch widerstandsfähige Schichten, von denen die eine als x- und die andere als y-Achse fungiert. Wenn Strom durch diese Schichten fließt, variiert ihr Widerstand je nachdem, wo sie angetippt wurden. Durch die Messung der Stromstärke in beiden Schichten lässt sich daher die x- und y-Koordinate des Berührungspunktes ermitteln.

In diesem Projekt erfassen wir mit dem Arduino die Stellen, an denen der Bildschirm berührt wurde, und wandeln die Messwerte in Integerzahlen um, die wir dazu verwenden können, auf die entsprechenden Flächen auf dem Bildschirm zu verweisen.

### Die Hardware

Sie benötigen folgende Hardware:

- Ein Adafruit Touchscreen, Teil 333
- Ein Adafruit Breakout-Board, Teil 3575
- Stecker-zu-Stecker-Überbrückungsdrähte
- Ein lötfreies Breadboard
- Arduino und USB-Kabel

Schließen Sie den Touchscreen nach der Anleitung in Tabelle 12–1 an. Verbinden Sie den Arduino über das USB-Kabel mit dem PC.

### Der Sketch

Geben Sie folgenden Sketch ein und laden Sie ihn hoch:

```
// Projekt 33: Bereiche auf dem Touchscreen ansprechen

#include <LiquidCrystal.h>
LiquidCrystal lcd(4,5,6,7,8,9);

int x,y = 0;

void setup()
{
  lcd.begin(16,2);
```

```

    lcd.clear();
}

int readX() // Gibt den Wert für die x-Koordinate der Berührung
           // zurück ❶
{
    int xr=0;
    pinMode(A0, INPUT);
    pinMode(A1, OUTPUT);
    pinMode(A2, INPUT);
    pinMode(A3, OUTPUT);
    digitalWrite(A1, LOW); // Setzt A1 auf GND
    digitalWrite(A3, HIGH); // Setzt A3 auf 5 V
    delay(5);
    xr=analogRead(0);      // Speichert den Wert der x-Koordinate
    return xr;
}

int readY() // Gibt den Wert für die x-Koordinate der Berührung zurück ❷
{
    int yr=0;
    pinMode(A0, OUTPUT); // A0
    pinMode(A1, INPUT);  // A1
    pinMode(A2, OUTPUT); // A2
    pinMode(A3, INPUT);  // A3
    digitalWrite(14, LOW); // Setzt A0 auf GND
    digitalWrite(16, HIGH); // Setzt A2 auf 5 V
    delay(5);
    yr=analogRead(1);     // Speichert den Wert der x-Koordinate
    return yr;
}

void loop()
{
    lcd.setCursor(0,0);
    lcd.print(" x = "); ❸
    x=readX();
    lcd.print(x);
    y=readY();
    lcd.setCursor(0,1);
    lcd.print(" y = "); ❹
    lcd.print(y);
    delay (200);
}

```

Die Funktionen `readX()` und `readY()` bei ❶ und ❷ lesen die Stromstärke in den Widerstandsschichten des Touchscreens mit `analogRead()` ab und geben den Wert zurück. Dieser Sketch führt diese beiden Funktionen rasch hintereinander aus, um die Position der berührten Fläche in Echtzeit zu ermitteln und bei ❸ und ❹ auf dem LCD-Bildschirm auszugeben. (Die Verzögerung `delay(5)` in den Funktionen ist erforderlich, damit die Ein-/Ausgangspins Zeit haben, ihren Zustand zu ändern.)

## Den Sketch testen

Um den Sketch zu testen, beobachten Sie den Serial Monitor, während Sie den Bildschirm berühren, und achten Sie darauf, wie sich die x- und y-Werte ändern, wenn Sie Ihren Finger auf dem Bildschirm bewegen. Achten Sie auch auf die Werte, die angezeigt werden, wenn der Bildschirm nicht berührt wird, wie in Abbildung 12–3 gezeigt.



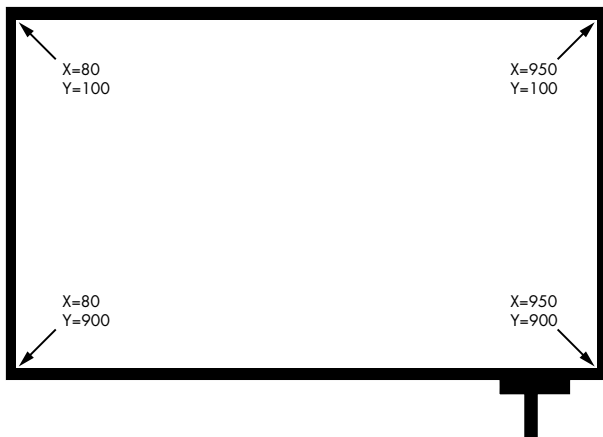
**Abb. 12–3** Werte bei nicht berührtem Touchscreen

Sie können die Werte, die angezeigt werden, wenn Sie den Bildschirm nicht berühren, in Ihrem Sketch verwenden, um zu erkennen, wann der Bildschirm nicht berührt wird. Außerdem können Displays leicht variieren. Daher ist es wichtig, dass Sie Ihr eigenes Gerät genau kalibrieren, damit Sie die Grenzen des Displays kennen.

## Den Touchscreen kalibrieren

Sie können die Koordinaten für jede Ecke Ihres Touchscreens ermitteln, indem Sie die Ecken berühren und die zurückgegebenen Werte festhalten, wie in Abbildung 12–4 gezeigt.

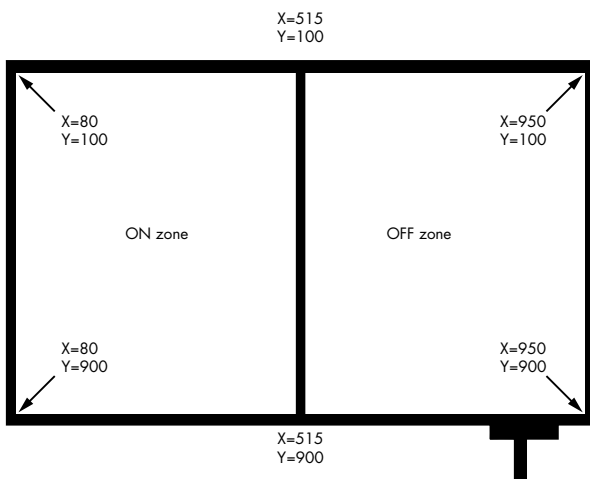
Nachdem Sie Ihre Touchscreen-Zuordnung erstellt haben, können Sie sie rechnerisch in kleinere Bereiche unterteilen, die Sie dann mit `if`-Anweisungen verwenden können, um bestimmte Aktionen auszulösen, je nachdem, wo der Bildschirm berührt wird. Das werden wir in Projekt 34 tun.



**Abb. 12-4** Zuordnung für einen Touchscreen

## Projekt Nr. 34: Einen Ein/Aus-Schalter mit zwei Zonen entwerfen

In diesem Projekt verwenden wir unsere Touchscreen-Zuordnung, um einen Ein/Aus-Schalter zu entwickeln. Als erstes teilen Sie den Touchscreen vertikal in zwei Hälften, wie in Abbildung 12-5 gezeigt.



**Abb. 12-5** Karte für den Ein/Aus-Schalter

Der Arduino bestimmt, welcher Bereich des Bildschirms berührt wurde, indem er die aufgezeichneten Koordinaten der Berührung mit den Grenzen der jeweiligen Bildschirmhälfte vergleicht. Wenn die Zone bestimmt wurde,