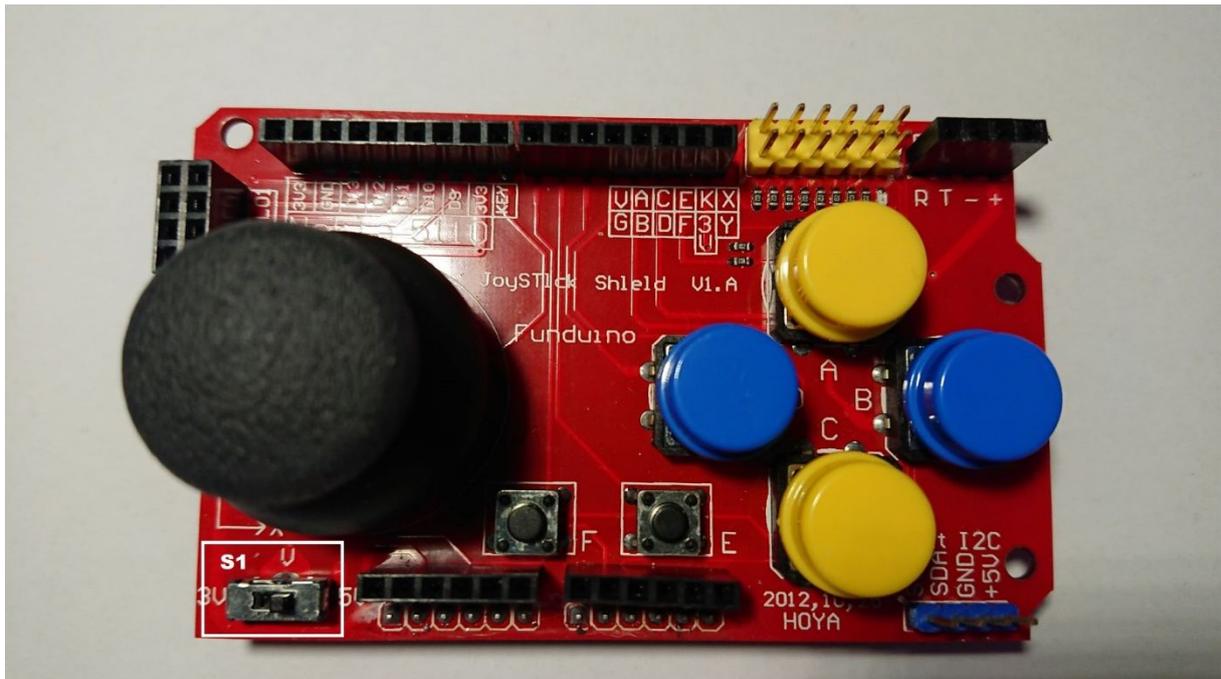


## Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery PS2 Joystick Shields. Auf den folgenden Seiten gehen wir mit dir gemeinsam den Aufbau und die ersten Programmierschritte durch.

Viel Spaß!



(Bild 1)

Der PS2 Joystick Shield (Funduino) ist ein Steck-Aufsatz für den Arduino Uno oder jedes andere Arduino Uno Spannungs- und Port kompatible Board.

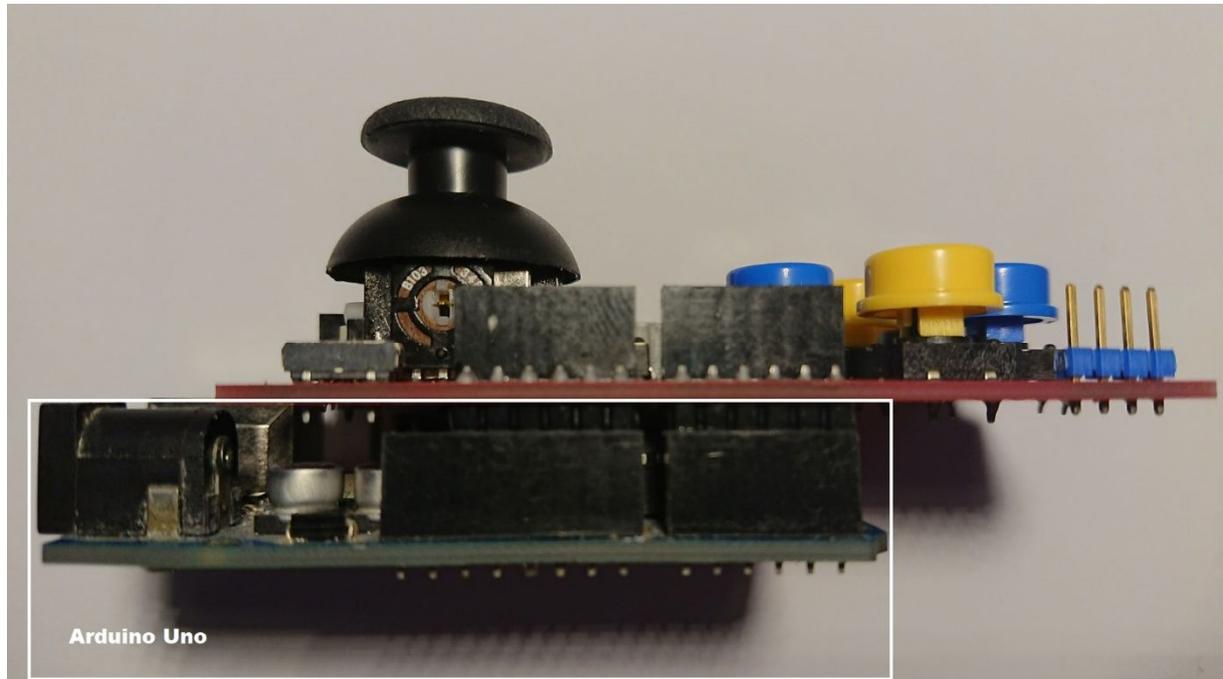
Es bietet digitales Steuerkreuz mit 4 Tastern, ein analoges 2 achsiges Steuerkreuz mit digitalem Drucktaster und zwei Zusatz Tastern zur erweiterten Steuerung.

Das PS2 Joystick Shield verfügt im Weiteren noch über verschiedene Anschlussmöglichkeiten für Erweiterungsmodule.

So ist es vorbereitet für den Anschluss eines 4 poligen Bluetooth Modules, eines nrf24L01 Funkmodules, eines Nokia 5150 Displays oder zur Anbindung von IC2 Buselementen.

In der vorliegenden Beschreibung konzentrieren wir uns auf die Hauptfunktionen des Shields. Die Anbindung von externen Komponenten wird ausführlich in einem anderen E-Book beschrieben.

Um mit dem Shield zu arbeiten, setze es bitte so wie hier gezeigt auf deinen Arduino auf:



(Bild 2)

Verbinde nun deinen Arduino mit deinem PC. Die rote Led auf deinem PS2 Joystick Shield sollte nun leuchten. Damit hast du schon alle notwendigen technischen Verbindungen geschaffen, um das Shield in Programmen verwenden zu können.

Um dir das Verständnis der Programmierung des Shields zu erleichtern, haben wir das E-Book in 3 Teile gegliedert. Im ersten Teil beschäftigen wir uns um die Anbindung der Taster (Digitalteil) und im zweiten Teil um die Anbindung des Joysticks (Analogteil) an den Arduino.

Im dritten Teil werden wir in einem Sketch den Bedienungsstatus der Taster und des Joysticks als Textinformation auf der seriellen Schnittstelle ausgeben.

Die Ausgabe kann dann beliebig mit einem eigenen Programm auf PC Seite ausgewertet werden.



# Az-Delivery

Kommen wir im zweiten Teil nun zu dem Analogteil des PS2 Joystick Shields und damit zu dem vielseitig nutzbaren Analogjoystick.

Der Analogjoystick ist technisch über ein horizontales (X Achsen-) Potentiometer an Analog Port 0 des Arduinos und einem vertikalem (Y Achsen-) Potentiometer an Analog Port 1 des Arduinos realisiert. Die Bewegung des Joysticks im Raum wird in verschiedene Spannungen an den Potentiometern umgesetzt.

Ein Federmechanismus im Joystick setzt den Steuerknüppel bei Nichtbetätigung immer in die Nullstellung (Mitte) zurück.

Durch kontinuierlich erfolgende Abfragen der Analogwandler Ports 0 und 1 kann die Knüppelposition abgefragt werden.

Achtung:

**!! Fortlaufende Analogdigital Wandungen des Prozessors erzeugen durch interne Messfehler nicht immer die exakt gleichen Werte, trotz gleicher Stellung des Steuerknüppels.**

**So kann der zurückgegebene Wert in der Nullstellung zwischen 1 bis 5 Werten schwanken. !!**

Durch einen Schalter S 1 (Bild 1) besteht die Umschaltmöglichkeit der Spannung von 5.5 Volt auf 3,3 Volt für die Spannungsteiler des analogen Steuerkreuzes.

Spannung (Schalterstellung S1)	Wertbereich:
5 Volt	0 ..1023
3,3 Volt	0..662

Auf diese Weise kann die Auflösung des analogen Joysticks geändert werden. Bei Einstellung auf 3,3 Volt ist Auflösung gemäß obenstehender Tabelle 663 Schritte und bei 5 Volt 1024 Schritte.

# Az-Delivery

Im letzten Abschnitt unseres E-Books kommen wir jetzt zu dem Arduino Sketch der die Daten des PS2 Joystick Shields über die serielle Schnittstelle des Arduinos ausgibt:

```
// Digital Eingänge, verbunden mit Tastern
const byte PIN_BUTTON_A = 2;
const byte PIN_BUTTON_B = 3;
const byte PIN_BUTTON_C = 4;
const byte PIN_BUTTON_D = 5;
const byte PIN_BUTTON_E = 6;
const byte PIN_BUTTON_F = 7;
const byte PIN_BUTTON_K = 8;

// Analog Eingänge, verbunden mit dem Joystick
const byte PIN_ANALOG_X = 0; // X Koordinate wird über Analogport o abgefragt Min:0 / Max: 1023
const byte PIN_ANALOG_Y = 1; // y Koordinate wird über Analogport o abgefragt Min:0 / Max: 1023

byte ButtonStatus = 0;
byte ButtonOLDStatus = 0;
byte JoystickButtonStatus = 1;
byte JoystickButtonOLDStatus = 1;

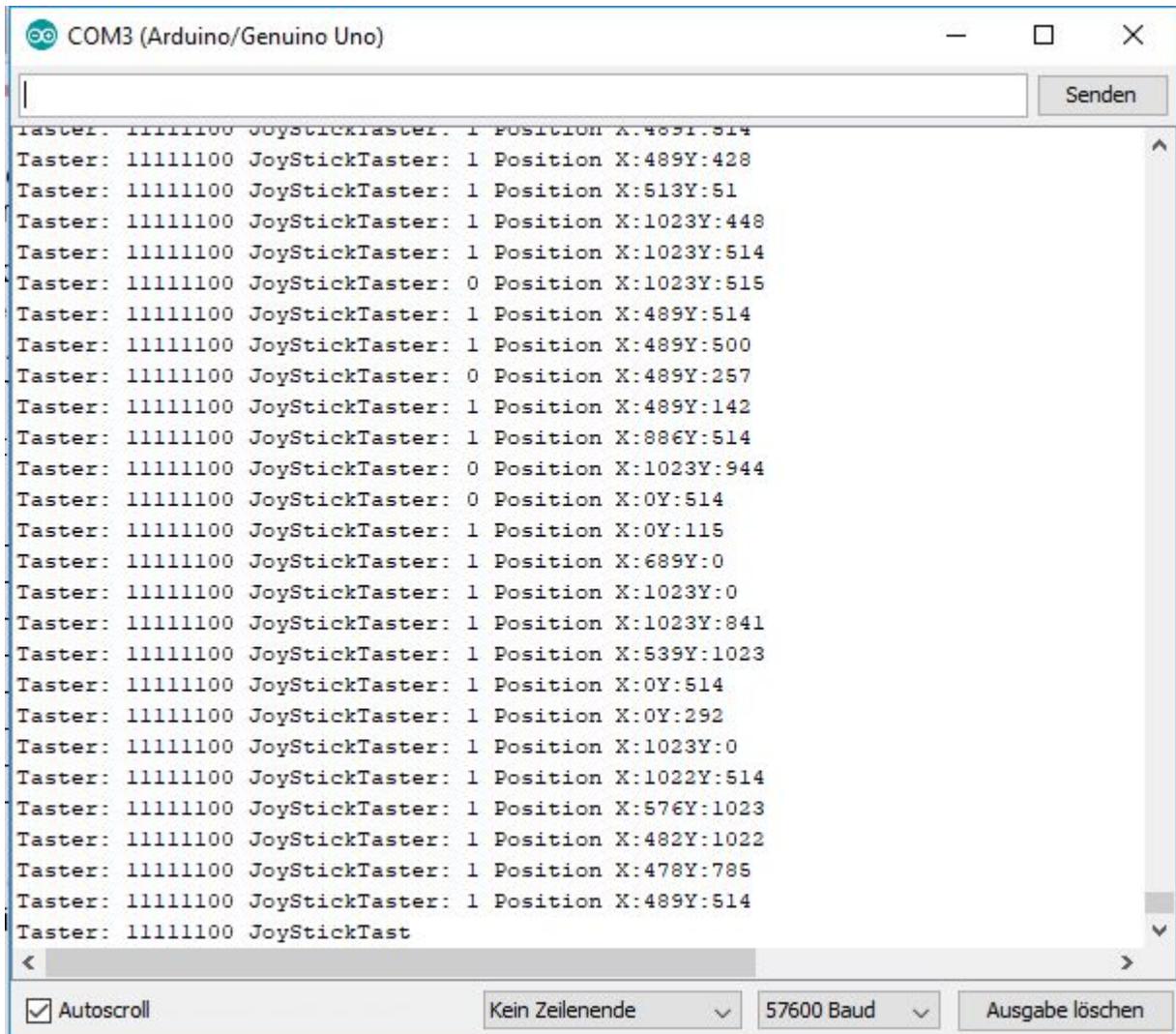
int XCoord = 0;
int XOLDCoord = 0;
int YCoord = 0;
int YOLDCoord = 0;

void setup() {
  Serial.begin(57600);
  pinMode(PIN_BUTTON_A, INPUT_PULLUP); //Pin 2 wird auf Eingang mit Pullup Widerstand konfiguriert. (Port D)
  pinMode(PIN_BUTTON_B, INPUT_PULLUP); //Pin 3 wird auf Eingang mit Pullup Widerstand konfiguriert. (Port D)
  pinMode(PIN_BUTTON_C, INPUT_PULLUP); //Pin 4 wird auf Eingang mit Pullup Widerstand konfiguriert. (Port D)
  pinMode(PIN_BUTTON_D, INPUT_PULLUP); //Pin 5 wird auf Eingang mit Pullup Widerstand konfiguriert. (Port D)
  pinMode(PIN_BUTTON_E, INPUT_PULLUP); //Pin 6 wird auf Eingang mit Pullup Widerstand konfiguriert. (Port D)
  pinMode(PIN_BUTTON_F, INPUT_PULLUP); //Pin 7 wird auf Eingang mit Pullup Widerstand konfiguriert. (Port D)
  pinMode(PIN_BUTTON_K, INPUT_PULLUP); //Pin 8 wird auf Eingang mit Pullup Widerstand konfiguriert.
}

void loop() {
  ButtonStatus = PIND & 0b11111100; //Einlesen der Eingänge 2-7 direkt über den Port D.
  // Eine bitweise UND Maske verhindert das Einlesen der Pins 0 und 1
  // (Serielle Schnittstelle)
  JoystickButtonStatus = digitalRead(PIN_BUTTON_K);
  XCoord = analogRead(PIN_ANALOG_X);
  YCoord = analogRead(PIN_ANALOG_Y);
  if ( (ButtonStatus != ButtonOLDStatus) || ( XCoord != XOLDCoord ) || ( YCoord != YOLDCoord ) || ( JoystickButtonStatus != JoystickButtonOLDStatus ) )
  {
    delay(100); // Taster Entprellen
    ButtonOLDStatus = ButtonStatus;
    JoystickButtonOLDStatus = JoystickButtonStatus;
    XOLDCoord = XCoord ;
    YOLDCoord = YCoord ;
    Serial.print("Taster: ");
    Serial.print(ButtonStatus,BIN);
    Serial.print(" JoyStickTaster: ");
    Serial.print(JoystickButtonStatus,BIN);
    Serial.print(" Position X:");
    Serial.print(XCoord);
    Serial.print("Y:");
    Serial.println(YCoord);
  }
  delay(10);
}
```

# Az-Delivery

Unser Programm erzeugt auf der Seriellen Schnittstelle bei 57600 Baud und S1 Schalterstellung 5Volt folgende Ausgabe:



```
Taster: 11111100 JoyStickTaster: 1 Position X:489Y:514
Taster: 11111100 JoyStickTaster: 1 Position X:489Y:428
Taster: 11111100 JoyStickTaster: 1 Position X:513Y:51
Taster: 11111100 JoyStickTaster: 1 Position X:1023Y:448
Taster: 11111100 JoyStickTaster: 1 Position X:1023Y:514
Taster: 11111100 JoyStickTaster: 0 Position X:1023Y:515
Taster: 11111100 JoyStickTaster: 1 Position X:489Y:514
Taster: 11111100 JoyStickTaster: 1 Position X:489Y:500
Taster: 11111100 JoyStickTaster: 0 Position X:489Y:257
Taster: 11111100 JoyStickTaster: 1 Position X:489Y:142
Taster: 11111100 JoyStickTaster: 1 Position X:886Y:514
Taster: 11111100 JoyStickTaster: 0 Position X:1023Y:944
Taster: 11111100 JoyStickTaster: 0 Position X:0Y:514
Taster: 11111100 JoyStickTaster: 1 Position X:0Y:115
Taster: 11111100 JoyStickTaster: 1 Position X:689Y:0
Taster: 11111100 JoyStickTaster: 1 Position X:1023Y:0
Taster: 11111100 JoyStickTaster: 1 Position X:1023Y:841
Taster: 11111100 JoyStickTaster: 1 Position X:539Y:1023
Taster: 11111100 JoyStickTaster: 1 Position X:0Y:514
Taster: 11111100 JoyStickTaster: 1 Position X:0Y:292
Taster: 11111100 JoyStickTaster: 1 Position X:1023Y:0
Taster: 11111100 JoyStickTaster: 1 Position X:1022Y:514
Taster: 11111100 JoyStickTaster: 1 Position X:576Y:1023
Taster: 11111100 JoyStickTaster: 1 Position X:482Y:1022
Taster: 11111100 JoyStickTaster: 1 Position X:478Y:785
Taster: 11111100 JoyStickTaster: 1 Position X:489Y:514
Taster: 11111100 JoyStickTast
```

Du hast es geschafft, du kannst nun in deinen Projekten das PS2 Joystick Shield verwenden..

Ab jetzt heißt es Experimentieren.

Und für mehr Hardware sorgt natürlich dein Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!  
Impressum

<https://az-delivery.de/pages/about-us>