

Eine „Trauung“ zwischen Retro-Idee und moderner Technologie.

Der heutige Aufsatz beschreibt als Retro-Idee, wie die Pong-Platine mit dem Wii-Nunchuk-Kontroller verbunden werden kann. Diese Verbindung ist relative einfach umzusetzen, dank der I²C/TWI-Schnittstelle, die beide Geräte besitzen.

I²C wird als *I-Quadrat-C* gesprochen und definiert einen Standard für eine serielle Master-Slave-Daten-Bus Kommunikation. Der Name „I²C“ steht für **Inter-Integrated Circuit**. Diese Schnittstelle wurde von Philips entwickelt und wird häufig verwendet, um verschiedene Peripherieschaltungen mit einem Haupt- μ -Controller mit geringer Übertragungsgeschwindigkeit (bis 400kbit/s) verbinden zu können.

Obwohl der Name I²C selbst kein eingetragenes Markenzeichen von Philips/NXP Semiconductors ist, verwenden viele Hersteller die Bezeichnung **TWI** „Two-Wire Interface“ (! nur das Logo ist nicht frei). Hinter beiden Namen: TWI und I²C verbirgt sich aber ein und derselbe Standard.

Der Kontroller

Den Nunchuk-Kontroller habe ich nicht grundlos ausgewählt. Dieses kompakte Gerät bietet für Steuerungszwecke viele interessante Funktionen. Es verfügt über einen Analog-Joystick, zwei Tasten und Beschleunigungssensoren. Für einen μ -Controller bzw. für unsere Pong-Platine stehen die Ausgabedaten der Sensoren (welche die Bewegungen des Nunchuks oder Joystick registrieren) schon in einer digitalisierten Form über die TWI-Schnittstelle zur Verfügung. Für den Status der zwei Tasten, des Joysticks und des 3-Achsen-Bewegungssensors werden nur zwei digitale Pins des Atmega8s genutzt und das ganze ist inklusive eines tollen Gehäuses für unter 15€ zu kaufen. Das ist weniger als für eine entsprechende Einzelbeschleunigungsschaltung.

Der I²C Pegelwandler

Leider ist die Welt nicht perfekt, und es gibt ein kleines Hindernis zu überwinden. Die Pong-Platine wird mit 4,5-5V betrieben und der Nunchuk-Kontroller aber nur mit 3,3V. Auf vielen Internet-Seiten wird zwar bestätigt, dass der Kontroller auch mit 5V arbeiten kann, aber aus meiner Sicht ist das nicht zu empfehlen, insbesondere, da der Hersteller des Beschleunigungssensors (im Nunchuk) für die Schaltung LIS3L02AL¹⁾ (das Herz des Geräts) nur eine Betriebsspannung von 2,4 bis 3,6V vorgesehen hat. Es gibt viele Lösungen von einfachen Pull-Up-Widerständen bis hin zu integrierten Schaltungen. Hier werde ich eine Lösung²⁾ des I²C-Entwicklers (Philips/NXP Semiconductors) bevorzugen, die in der „Application Note“ AN97055 als ein bidirektionaler Pegelwandler 5V/3,3V für I²C-Systeme veröffentlicht wurde. Diese Lösung habe ich unten auf dem Bild 1 mit entsprechenden Elementen und einem Spannungsregler in der Leitung dargestellt.

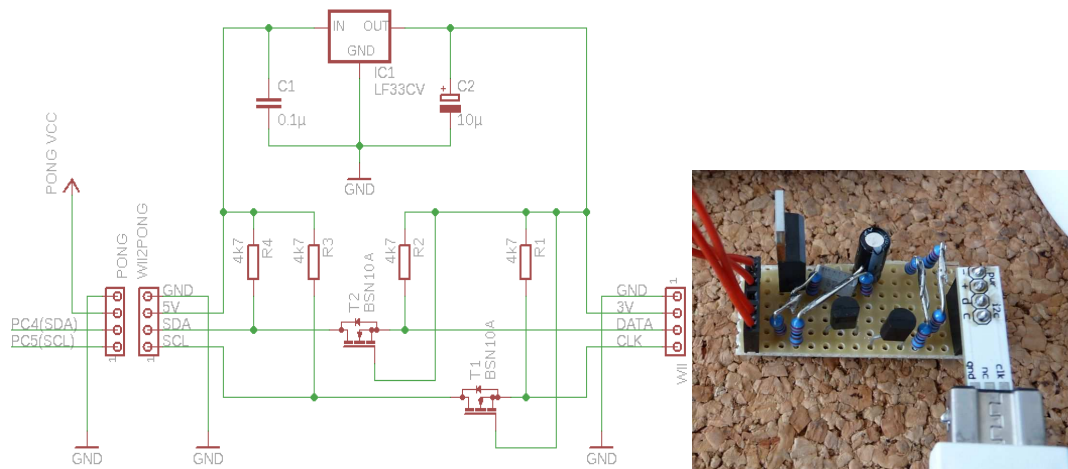


Bild 1: Der I²C Pegelwandler

Jetzt kann der Nunchuk endlich an der Pong-Platine angeschlossen werden. Für eine direkte Verbindung könnte der Nunchuk-Stecker einfach abgeschnitten werden - Achtung: die Familie kann protestieren. Aber es gibt auch eine andere Lösung. Im Internet gibt es einen „Wiichuck Adapter“³⁾ oder „Extension Cable for Wii Nunchuk“⁴⁾ für wenige Euro zu kaufen.



Bild 3: Wiichuck Adapter

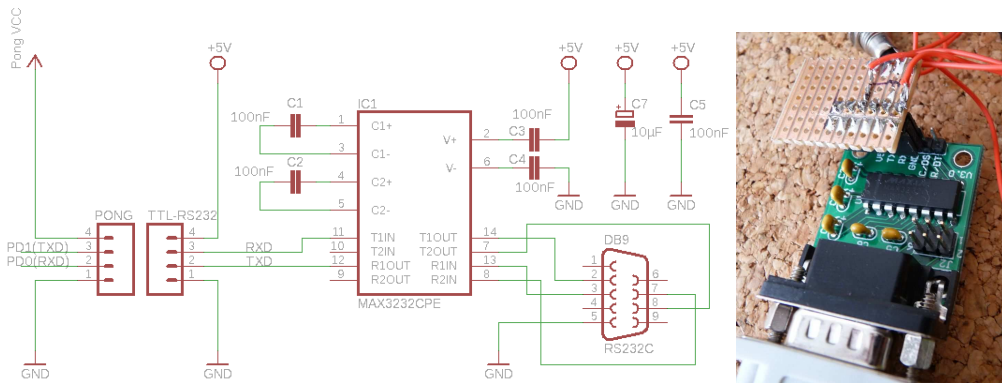


Bild 2: Extension Cable

Der Atmega8 unterstützt die „Two Wire Serial-Schnittstelle“ schon auf der Hardwareebene. Damit ist der Softwareteil relativ einfach. Für eine erste „Kennenlernrunde“ habe ich BASCOM genutzt. Die BASCOM-Sprache vereinfacht die Initialisierung und das Ablesen der Schnittstellen. Das Programm „pongunchuk.bas“ initialisiert die TWI des Atmega8s, das Nunchuk-Slave-Gerät und liest regelmäßig die ankommenden Daten ein. Die Daten werden binärkodiert auf dem LED-Display angezeigt. Es blinkt - so weit so gut! Um die Daten vom Nunchuk in einem freundlicheren (als binärkodiert) Format darzustellen, brauchen wir entweder einen Bildschirm/Display, den die Pong-Platine nicht besitzt, oder wir schicken, wie in der µ-Controller-Welt üblich, die Datenreihe an einen PC.

Der TTL/RS232C Pegelwandler

Der kleine Atmega8 kann uns auch hier mit seiner seriellen Schnittstelle voll unterstützen. Es wird wieder nur ein Pegelwandler zwischen TTL (5V) und RS232C (12V) benötigt. Glücklicherweise kann diese Schaltung bei Ebay oder im Hobbyladen ohne Probleme gekauft werden. Einfach nach „TTL RS232C Wandler“⁵⁾ suchen. Obwohl



der Wandler nicht notwendig ist, um den Nunchuck an die Pong-Platine anzuschließen, ist eine RS232C-Verbindungsmöglichkeit für jeden Pong-Bastler sehr zu empfehlen. Mein RS232C-Pegelwandler (für 6€) und die Schaltung werden unten gezeigt. Nun kann die Schaltung mit der Pong-Platine und dem PC über ein RS232C-Kabelmodem (das gleiche Kabel, wie für die Programmierung des Pongs) am anderen Ende verbunden werden. Auf dem Computer muss ein Terminal-Programm, z.B. Windows HyperTerminal, gestartet werden (die Hardcopy in Bild 5 zeigt die Verbindungsparameter). Nach Anlegen einer Betriebsspannung an die Pong-Platine fließen die Daten auf den PC-Bildschirm.

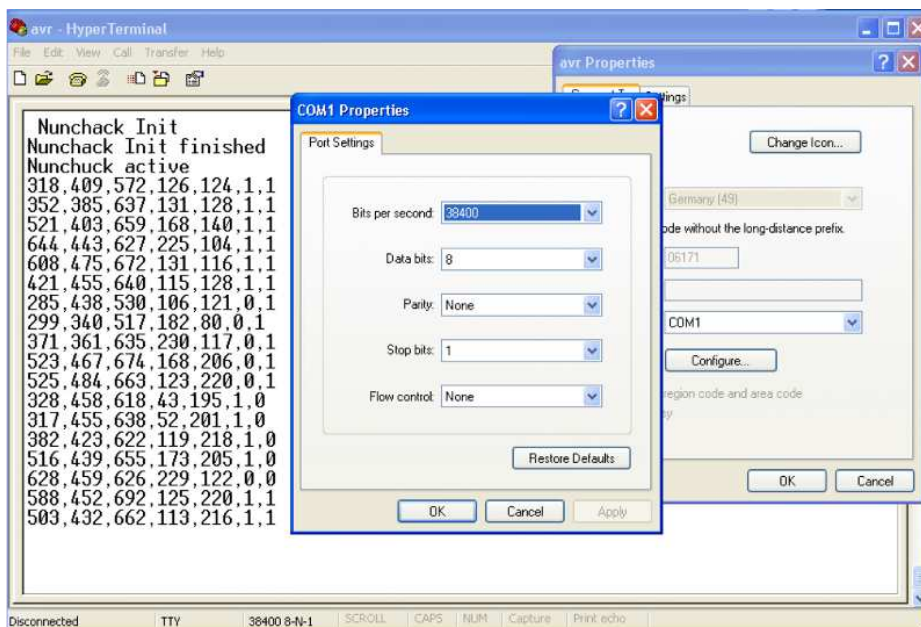


Bild 5: Verbindungsparametern

Ein Versuch in C und eine Einführung in „Processing“⁶⁾

Die große Menge an vorhandenen Daten lässt sich nicht einfach beherrschen. Um ein Spiel bzw. ein Programm, das den Nunchuk nutzt, entwickeln zu können, ist vorab Kenntnis über das Datenverhalten (vom Nunchuk) und die mögliche Bandbreite nötig. Das Verhalten von vielen und sich schnell ändernden Daten, die maximalen und minimalen Werte können sehr gut über eine grafische Darstellung analysiert werden. Für diese und alle ähnlichen Aufgaben würde ich eher die „Processing“-Umgebung bevorzugen.

Die „Processing-Umgebung“ ist eine vereinfachte Java-basierte Programmiersprache mit begleitender, voll-integrierter Entwicklungsumgebung. Die Processing-Sprache und die vorhandenen Klassenbibliotheken unterstützen die umfangreiche Verarbeitung der Daten in den verschiedenen Bereichen: Grafik, Video, Sound, Netzwerkprotokolle und serielle Schnittellendaten. „Processing“ ist ein quell-offenes betriebssystemunabhängiges Projekt und kann kostenlos genutzt werden.

Die Verbindung zwischen Pong und Wii-Nunchuk hat mit dem BASOM-Programm schon gut funktioniert. In C sind Initialisierung und Bedienung der Schnittstellen zwar flexibler als in BASCOM, allerdings muss alles auf der niedrigsten Hardwareebene programmiert werden. Da die TWI-Schnittstelle sehr häufig genutzt wird, gibt es viele Beispiele und offene Quellen für die Initialisierung und Bedienung des Hardwareteils. Für diesen speziellen Einsatz des Pong-Wii-Nunchuks wird eine Bibliothek des Arduino-Projekts⁷⁾ angepasst. Die Aufgabe des neuen Testprogramms ist die Umwandlung der Rohdaten aus der Schräglage des Beschleunigungssensors des Nunchuks in Winkeldaten. Des Weiteren sollen die Daten des Joysticks in „signed byte“-Kettenwerte umgerechnet werden. Dafür werden die Funktionen „nunchuckRead“ und „normalize“ mit entsprechenden Maximal- und Minimalrohdaten verwendet. Die Bandbreite (Min-/Max-Werte) der Rohdaten wurde mit Hilfe des Processings ermittelt. Ein fertiges C-Testprogramm „pongwii-test.zip“ und einen Processing-Sketch „pong_wii_nuchuk_testing.pde“ ist hier zur Verfügung gestellt. Die Ausgabedaten des Testprogramms können auch mit dem RS-Terminal angeschaut werden. Das Bild 6 stellt das laufende Processing für das Pong-Wii-Testprogramm dar. Ein autonom ausführbarer Processing-Sketch (ohne Entwicklungsumgebung: pong_wii_nuchuk_testing\application.windows\pong_wii_nuchuk_testing.exe) ist auch vorhanden.

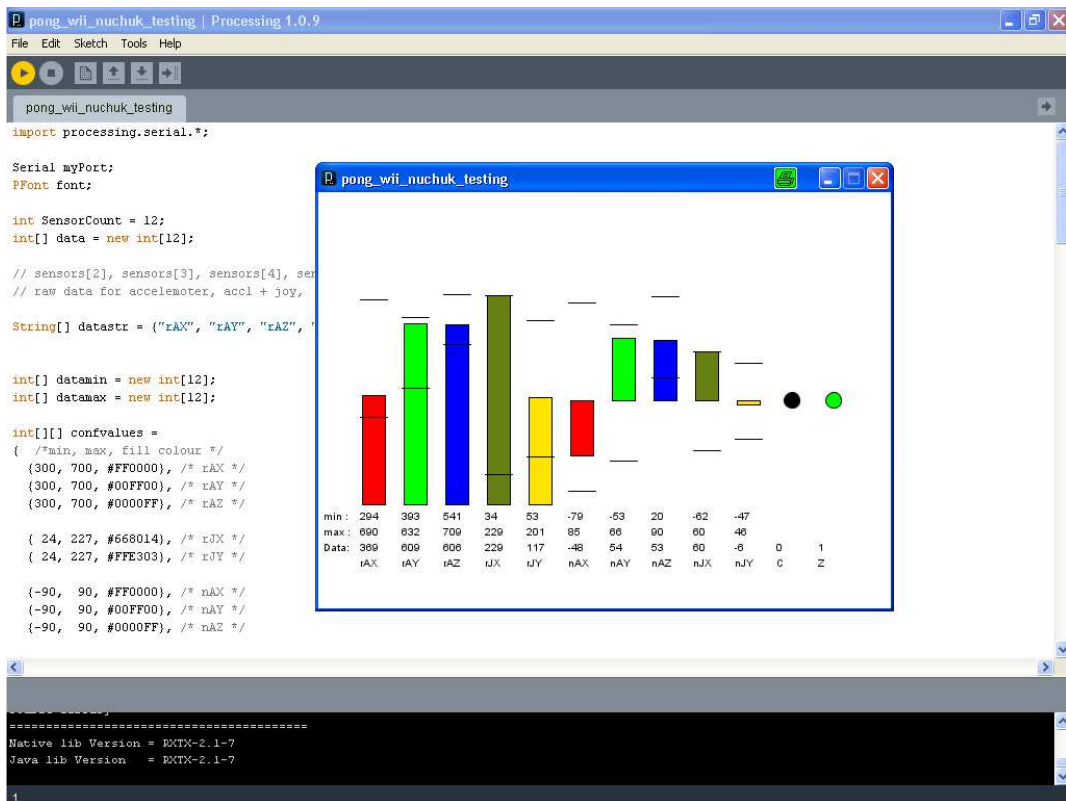


Bild 6: Processing - Test des Nunchuks

Das Spielkonzept

Hardware- und Softwareteil wurden umfassend getestet. Jetzt ist die Zeit dieses Wissen und die Hardware in einem Spiel anzuwenden. Lassen Sie uns die Pong-Platine als ein Holzbrett mit einem Stahlkugel betrachten. Die symbolische Kugel rollt hin und her entsprechend der Neigung des Bretts. Das Ziel des Spiels ist die Verfolgung eines Punktes, der nach einer gewissen Zeit wieder verschwindet. Der Spieler verfügt nur über eine begrenzte Zeitspanne (z.B. einige Sekunden) bis der Zielpunkt verschwindet und an anderer Stelle des Brett wieder auftaucht. Die Kugel darf nicht vom Brett fallen. Anfangs hilft dem Spieler ein symbolischer Zaun, damit die Kugel auf dem Brett bleibt. „Leider“ wird der Zaun im Laufe des Spiels durch Explosionen, immer dann wenn der Zielpunkt getroffen wird, nach und nach zerstört, so dass die Kugel leichter vom Brett fallen kann.

Die Neigung des Bretts wird über den Beschleunigungssensor gemessen. Die Kugel wird mittels einer permanent leuchtenden LED dargestellt. Im Gegensatz dazu blinkt der zu verfolgende Zielpunkt. Derjenige, der die meisten Zielpunkte trifft, gewinnt das Spiel. Am Ende des Spiels wird das Ergebnis angezeigt. Ein neues Spiel wird mit dem Knopf „C“ des Nunchuks gestartet. Das Konzept des Spiels wird auf dem Bild 7 symbolisch dargestellt.

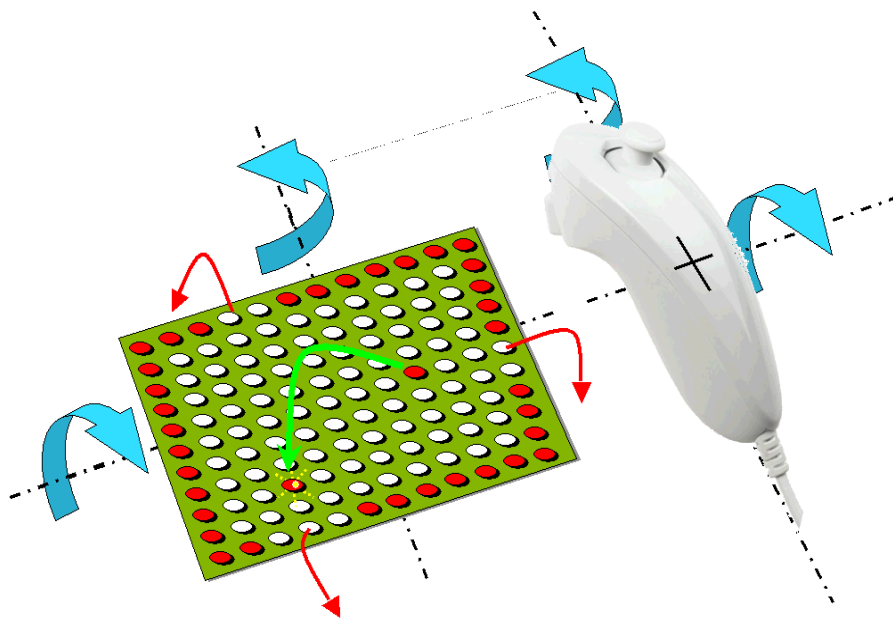


Bild 7: Spielkonzept

Die Pong-Platine kann mit ein Gummiband direkt auf dem Nunchuk befestigt werden. Der herausragende Joystick wird mit einer Tesafilmrolle in entsprechendem Abstand zur Platine gehalten. Als Streuscheibe dient hier zwei Lagen Pauspapier. Für ein professionelles Spiel sollte die integrierte Schaltung des Beschleunigungssensors direkt auf die Platine gelötet werden.

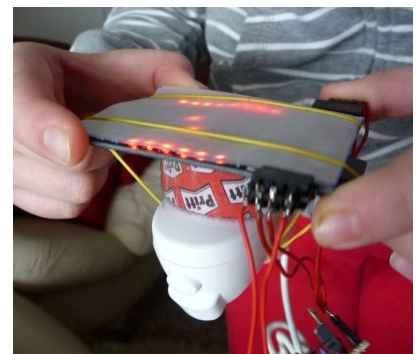
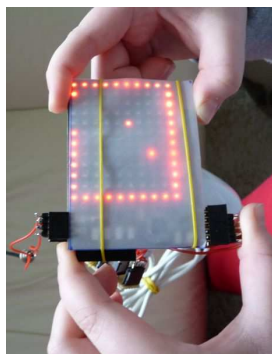
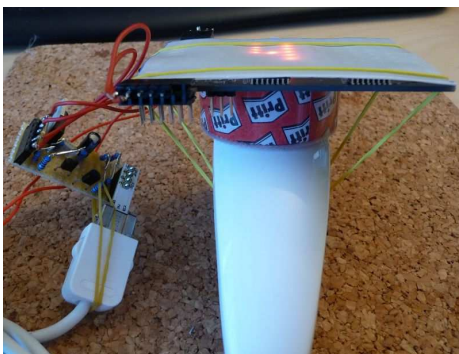


Bild 8: Galerie der Pong-Platine mit dem Nunchuk

Für ungeduldige Tester, die weder Nunchuk noch Zeit für den I²C Pegelwandler haben, wurde eine vereinfachte Version des Spiels (pong-hunt.zip) geschrieben. In dieser Version werden die Standard-Potis der Pong-Platine als „Beschleunigungssensor“ genutzt. Allerdings ist diese Version ziemlich schwer zu spielen und verliert definitiv den Charme der physischen Bewegungsnachbildung.

Die entsprechenden Programme liegen sowohl in C als AVR-Studio-Projekte, als auch in BASCOM und Processing-Sketch zum Download vor. Viel Spaß mit dem Aufbau und Spielen.

-
- 1) Beschleunigungssensor Schaltung LIS3L02AL: <http://www.st.com/stonline/products/literature/ds/11668/lis3l02al.htm>
 - 2) I²C Levelshifter: <http://ics.nxp.com/support/documents/interface/pdf/an97055.pdf>
(<http://ics.nxp.com/support/documents/>)

-
- 3) <http://store.fungizmos.com>
(http://store.fungizmos.com/index.php?main_page=product_info&cPath=69&products_id=212) oder
<http://www.watterott.com> (<http://www.watterott.com/de/Leiterplatten/WiiChuck-Adapter>)
 - 4) <http://ebay.de/> Suchen nach "Extension Cable for Wii Nunch*"
 - 5) TTL/RS3232 Levelshifter http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1068
 - 6) Processing: <http://processing.org/>
 - 7) Arduino: <http://www.arduino.cc/>