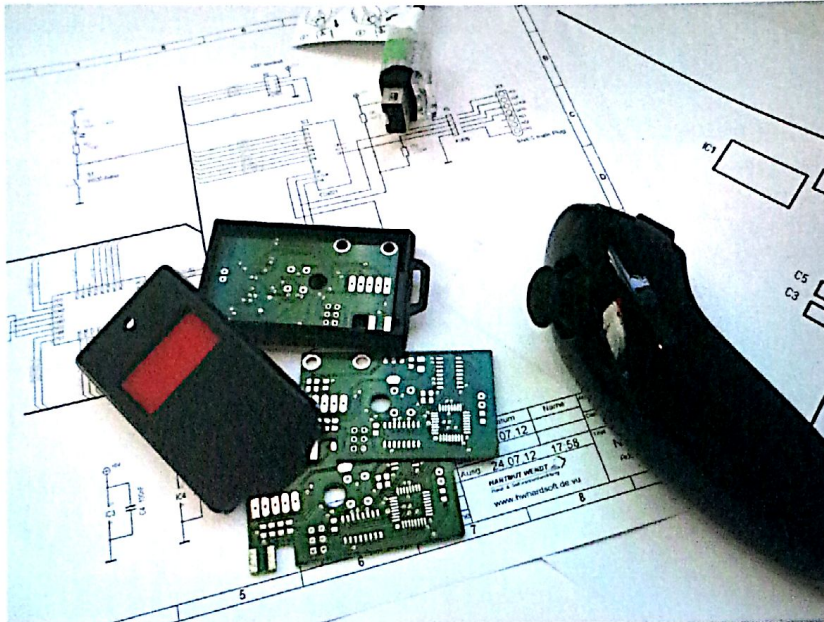


NUN2SNES: So könnt Ihr mit Wii-Fernbedienung und -Nunchuk auch Eure 8Bit- bzw. 16Bit-Konsolen bedienen.

# NUN2SNES:

## ein Nunchuk-Interface für (S)NES



Fein säuberlich in seine Einzelteile zerlegt: Das "NUN2SNES"

Als Nintendo 2006 die Wii auf den Markt brachte, stellten die neuen Controller Wii-Remote und der dazugehörige Wii-Nunchuk ein Novum dar. Bisher hatte man Spiele mit Hebeln und Knöpfen in Joysticks und Gamepads gesteuert. Doch nun war es plötzlich möglich, die Lage- und Positionsänderung des Controllers zu erfassen und somit ein viel freieres und intuitiveres Steuerungskonzept zu verwirklichen. Mittlerweile hat der Wettbewerb mit der Kinect bei der XBOX 360 oder dem Move Controller der Playstation dem sogar noch eins draufgesetzt. Eine ganze Generation von jungen Spielern ist bereits mit dieser Art der Spielsteuerung aufgewachsen und hat eher ein distanzierteres Verhältnis zu den früher gebräuchlichen Gamepads und Joysticks. In diesem Artikel möchte ich ein Interface zum Anschluss des Wii-Nunchuk-Controllers an die NES und SNES Konsole vorstellen (Bild oben). Mein Ziel bei dieser Entwicklung war es, Spieleklassiker dieser Konsolen auch für jüngere Spieler interessant zu machen.

### WII NUNCHUK

Um ein solches Interface zu bauen, muss man sich zunächst mit der Funktionsweise des Nunchuk-Controllers vertraut machen. Das

Und so sieht es von innen aus: Unser geöffnetes NUN2SNES im Querschnitt.



Nunchuk besteht aus drei Beschleunigungssensoren, die jeweils für eine Achse in Bezug zur Erdgravitation Lageänderungen feststellen können. Zusätzlich verfügt das Nunchuk noch über einen analogen Joystick auf der Oberseite und zwei unabhängige Feuerknöpfe. Die Daten dieser drei verschiedenen Eingabeelemente können über ein serielles Bussystem abgefragt



Das rundum verkabelte NUN2SNES von vorne.

werden. Hier hat sich Nintendo keine eigene Lösung ausgedacht, sondern verwendet den bekannten I2C-Bus der Firma Philips, der seit Jahren in elektronischen Geräten aller Art zur Anwendung kommt. Dadurch ist die Kommunikation mit dem Nunchuk keine Raketentechnik, sondern sehr leicht nachzuvollziehen. Die Daten für die Lage des Nunchuks, der Joystickposition und des Zustands der Feuerknöpfe werden in sechs Bytes übermittelt. Das Interface fragt nun in regelmäßigen Abständen den Zustand der Eingabeelemente ab.

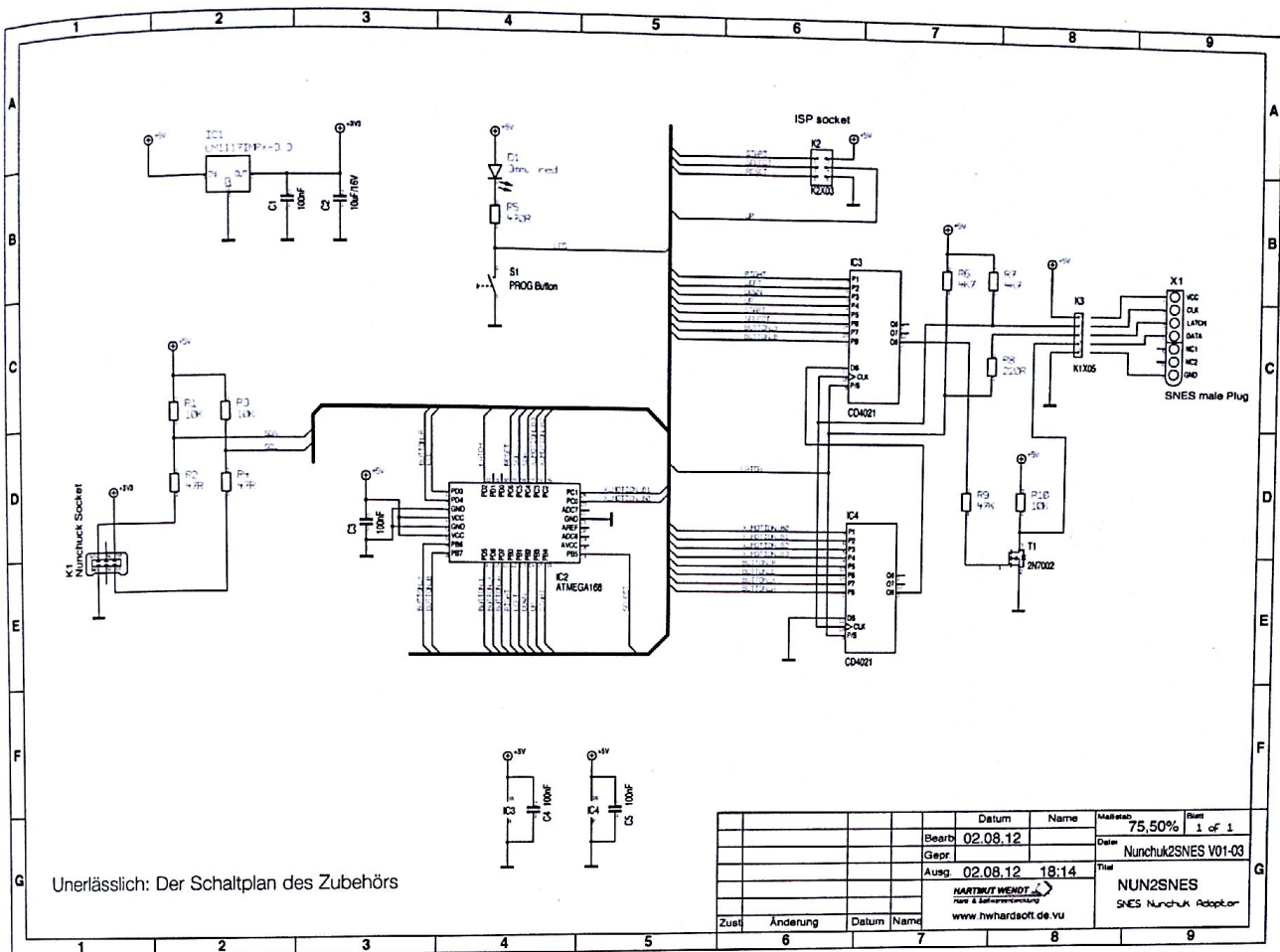
### NES- UND SNES-KOMMUNIKATION

Auch bei der Kommunikation mit den Gamepads von NES und SNES setzte Nintendo seinerzeit auf Bewährtes. Die NES Konsole nutzt zur Kommunikation mit dem Gamepad ein synchrones serielles Protokoll, das die drei Signale Clock, Daten und ein Latchsignal verwendet. Diese Schnittstelle entspricht dem bekannten Serial Peripheral Interface (kurz SPI) der Firma Motorola. Es werden acht Bit übertragen, die die Zustände der Feuertasten und des Steuerkreuzes enthalten. Bei dem SNES hat man dieses Prinzip beibehalten und lediglich die Information auf 16 Bit erweitert, um die Information der zusätzlichen Feuerknöpfe unterbringen zu können. Da diese Datenmenge für eine Maus, bei der ja Positionsdaten für X und Y notwendig sind, nicht ausreicht, wird bei der SNES-Maus dieses 16-Bit Protokoll einfach zweimal hintereinander gesendet. Die Konsole gibt dabei den Takt über die Clock-Leitung und den Zeitpunkt der Datenübernahme im Gamepad über das Latchsignal vor. Dieser Umstand sollte sich bei meinen Bemühungen noch als problematisch erweisen.

### DIE SCHALTUNG

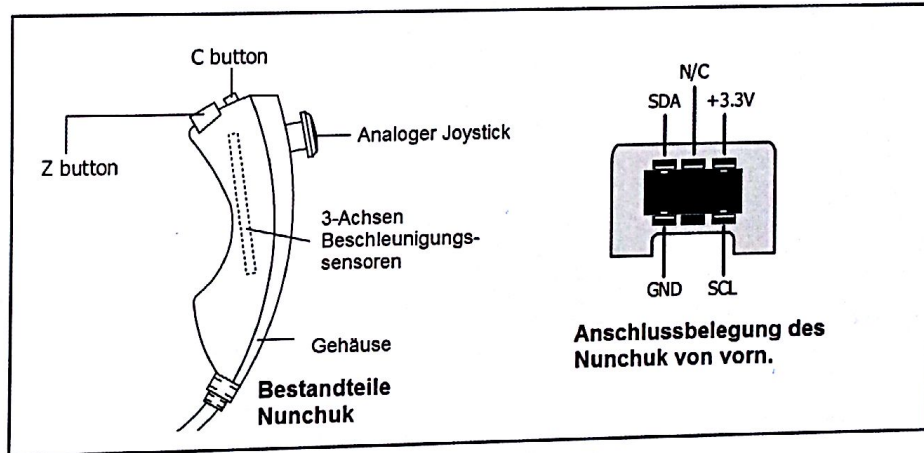
Das eigentliche Interface besteht aus einem 8Bit-Mikro-Controller ATMEGA168 der Firma ATMEL. Dieser Mikro-Controller besitzt eine





Unerlässlich: Der Schaltplan des Zubehörs

Feuerknöpfe	LED
C - Button A Z - Button B	LED immer an
C - Button A Z - Button B	LED alle 2s kurz aus
C - Button rechts Z - Button links	LED blinkt schnell
C - Button rechts Z - Button links	LED blinkt langsam



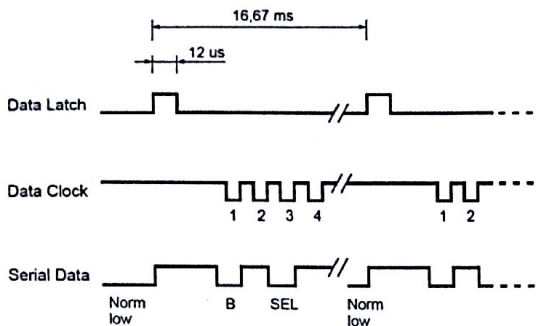
Das Nunchuk für Nintendos Wii-Konsole aus der Horizontalen: Diese Skizze schlüsselt die Konstruktion des innovativen Controllers auf.

Hardware I2C und auch SPI-Schnittstelle, was ihn eigentlich für diese Aufgabe prädestiniert. Die I2C Schnittstelle sollte als Master und die SPI-Schnittstelle als Slave (der Takt kommt vom NES bzw. SNES) betrieben werden. Allerdings zeigte sich bei den ersten Aufbauten auf einem Breadboard relativ schnell, dass insbesondere die Abfrage der 16 Bit durch das SNES mit dem Hardware-SPI des ATMEGA168 nicht realisierbar

ist, da hier nur eine Wortbreite von acht Bit unterstützt wird. Es dauert einfach zu lange, weitere acht Bit in das entsprechende Register im Mikro-Controller zu laden. Zähneknirschend musste ich deshalb die Schaltung des Interfaces um zwei serielle Schieberegister vom Typ 4021 erweitern. Diese beiden Schieberegister bilden die Elektronik eines SNES-Gamepads nach. Beim Anschluss an das NES wird einfach nur das erste Schieberegister genutzt. Die Eingänge der 4021 entsprechen dabei den zugeordneten

Bits des Übertragungsprotokolls. Eine weitere Besonderheit stellt die zusätzliche Auswertung des Latch-Signals durch den Mikro-Controller dar. Das Latch-Signal lädt im Normalfall automatisch den aktuellen Zustand der Eingänge in die Schieberegister. Allerdings können so immer nur 16 Bit übertragen werden. Zur Emulation der SNES-Maus müssen aber zwei 16 Bit-Protokolle mit unterschiedlichen Informationen hintereinander übertragen werden. Zunächst wird dazu der Anschluss PD2





Die Kurven zeigen die Übertragungs-Protokolle von NES- und SNES-Gamepad.

Clock	NES Gamepad	SNES Gamepad	SNES Maus
1	A	B	high
2	B	Y	high
3	Select	Select	high
4	Start	Start	high
5	Up	Up	high
6	Down	Down	high
7	Left	Left	high
8	Right	Right	high
9		A	Left Button
10		X	Right Button
11		L	high
12		R	high
13		high	high
14		high	high
15		high	high
16		high	low
17			Y Richtung
18			Y motion bit 6
19			Y motion bit 5
20			Y motion bit 4
21			Y motion bit 3
22			Y motion bit 2
23			Y motion bit 1
24			Y motion bit 0
25			X Richtung
26			X motion bit 6
27			X motion bit 5
28			X motion bit 4
29			X motion bit 3
30			X motion bit 2
31			X motion bit 1
32			X motion bit 0

OBEN: Bitzuordnung im (S)NES Übertragungsprotokoll

UNTEN: Übersicht der Betriebsarten

Betriebsart	Konsole	Kippen	Joystick
Mode 1 - Gamepad Emulation mit Beschleunigungssensoren	NES & SNES	Richtungssteuerung	Links - SHLD L Rechts - SHLD R Hoch - START Runter - SEL
Mode 2 - Gamepad Emulation des Joystick	NES & SNES	Links - SHLD L Rechts - SHLD R Hoch - START Runter - SEL	Richtungssteuerung
Mode 3 - Mouse Emulation mit Beschleunigungssensoren	nur SNES	Mauszeiger	
Mode 4 - Mouse Emulation mit Joystick	nur SNES		Mauszeiger

des Mikro-Controllers als Eingang mit Pinchange-Interrupt betrieben. So kann der Latch-Impuls des SNES vom Mikro-Controller detektiert werden. Danach schaltet der Mikro-Controller PD2 als Ausgang. Nach dem Senden der ersten 16 Bit des Protokolls stellt der Mikro-Controller nun die oberen 16 Bit der Maus-Information an den Eingängen des Schieberegisters zur Verfügung und erzeugt selbst via PD2 einen weiteren Latch-Impuls, der diese Information in die beiden Schieberegister lädt.

**FUNKTIONSWEISE**

Das Interface unterstützt je nach Konsole zwei bzw. vier unterschiedliche Betriebsarten. Generell kann beim SNES zwischen der Emulation des Gamepads oder der Maus gewählt werden. Für das NES steht nur die Emulation des Gamepads zur Verfügung. Weiterhin kann die Betätigung des Steuerkreuzes am Gamepad bzw. die Bewegung der Maus entweder durch die integrierten Beschleunigungssensoren des Nunchuks, also das Kippen des Nunchuks nach links, rechts, vorn und hinten oder durch Betätigen des Joysticks an der Oberseite des Nunchuks simuliert werden. Die Neigung des Nunchuks oder die Auslenkung des Joysticks wird dabei nicht einfach 'digital' in eine Betätigung des Steuerkreuzes übertragen, sondern quasi 'analog' durch eine unterschiedliche Wiederholrate der Tastenbetätigung umgesetzt.

Die beiden Buttons des Nunchuks entsprechen den Feuerknöpfen A und B des Gamepads bzw. den beiden Maustasten. Weitere Tasten des Gamepads können durch Kippen des Nunchuks oder Bewegen des Joysticks simuliert werden.

Die gewünschte Betriebsart hängt vom verwendeten Spiel ab. Eine Maus-Emulation ist natürlich nur für Spiele sinnvoll, die auch eine Maus unterstützen. Ob es besser ist, die Steuerung mit dem Joystick oder durch Kippen des Nunchuks durchzuführen, muss von Spiel zu Spiel ausprobiert und entschieden werden. In jedem Fall lässt sich die Betriebsart mit dem Programmierertaster S1 auswählen. Dazu wird S1 für mindestens eine Sekunde gedrückt. Die LED leuchtet in dieser Zeit dauerhaft. Nach dem Loslassen des Programmierertasters wird die aktuelle Betriebsart durch unterschiedliches Blinken der LED angezeigt. Mit jeder Betätigung des Programmierertasters wird immer die nächstfolgende Betriebsart gewählt. Eventuell muss dieser Vorgang also mehrfach wiederholt werden, bis die gewünschte Betriebsart eingestellt ist.

**AUFBAU UND TIPPS**

Im einfachsten Fall kann die Schaltung auf einer Lochrasterkarte nachgebaut werden. Die erforderlichen Teile sind im Elektronikfachhandel leicht erhältlich. Ein passendes Kabel zum Anschluss an NES oder SNES kann man kostengünstig durch Ausschachten eines entsprechenden (Billig-)Gamepads gewinnen.

Ich habe für den Aufbau von mehreren Prototypen ein eigenes Layout erstellt. Dieses Layout wurde so optimiert, dass die resultierende Platine in ein kleines Kunststoffgehäuse von der Größe einer Streichholzschachtel passt. Das Nunchuk kann über eine seitliche Öffnung im Gehäuse auf eine passende Zunge in der Leiterplatte gesteckt werden, die die erforderlichen Kontakte bereitstellt. Das Kabel zur Konsole wird an die Platine angelötet und für die Zugentlastung über eine Schelle mit der Platine verschraubt.

Weiterführende Informationen sind auf der Website des Autors [www.hwhardsoft.de](http://www.hwhardsoft.de) zu finden. Hier steht auch die Firmware für den Mikro-Controller zum Download bereit.

Das Interface wurde bisher nur mit Original-Zubehör von Nintendo getestet. Erfahrungen zu Nunchuk-Kopien von anderen Herstellern liegen nicht vor. Die Firmware wird aber sukzessive immer weiter verbessert. (hw)