

Datenhandschuh

Dokumentation eines Projektes

6. März 2011

Inhaltsverzeichnis

I	Einführung	5
1	Vorwort	7
2	Abstrakt	9
II	Hardware	11
3	Übersicht der Hardware	13
4	Einzelne Teile der Hardware	15
4.1	Aufnahme der Messdaten	15
4.2	Beschaltung und Wechsel	15
4.3	AD-Wandler	16
III	Software	19
5	Übersicht	21
6	Datenaufnahme	23
6.1	Serielle Schnittstellen unter Linux	23
6.2	Datensammlungsmethode	23
7	Neuronale Netze	25
7.1	Einführung in Neuronale Netze	25
7.2	Verwendetes Neuronales Netz	25
7.3	Umsetzung	25
IV	Erfolge und Probleme	27
V	Abschluss	29
	Index	31

Teil I

Einführung

Kapitel 1

Vorwort

Dieses Dokument beschreibt die Kenndaten und Hauptinhalte eines Projektes, das von mir Anfang 2011 begonnen wurde. Es sollen die Ideen und von mir genutzten Umsetzungsmöglichkeiten aufgezeigt und die aufgetretenen Probleme besprochen werden. In keinem Fall handelt es sich hierbei um eine komplette Einführung in die besprochenen Themengebiete oder eine nachvollziehbare Anleitung zum Nachbau des Datenhandschuhs. Vielmehr soll dieses Dokument die reine Möglichkeit und Umsetzbarkeit eines solchen Projektes aufzeigen.

Bruchsal, 6. März 2011
Nils Braun

Kapitel 2

Abstrakt

Der Datenhandschuh besteht aus mehreren Komponenten. Auf Hardwareseite stehen da vor allem die fünf Beschleunigungssensoren, von denen vier die Fingerkuppenbewegungen (außer dem kleinen Finger) aufnehmen und einer die ganze Handbewegung. Die analogen Spannungsdaten werden über einen AD-Wandler seriell an den Computer gesendet. Dort, durch die serielle Schnittstelle aufgenommen, gelangen die Daten zu einem vorher trainierten Neuronalen Netz und werden von diesem kategorisiert. Somit lassen sich einzelne Gesten voneinander unterscheiden und somit auf Benutzereingaben reagieren. Die ganze Bandbreite der möglichen Interaktion des Benutzers mit dem Computer - wie sie durch tastatur und Maus erreicht wird - lässt sich somit natürlich nicht abdecken. Jedoch bietet diese Art der Kommunikation eine Möglichkeit, auf ganz andere Aspekte der Ausdrucksformen einzugehen.

Teil II

Hardware

Übersicht der Hardware

Wie auf diesem Schaltbild (Abb. **3.1**) zu sehen ist, lässt sich die verwendete Hardware grob in drei Teilaspekte einteilen.

- Aufnahme der Messdaten über die Sensoren
- Beschaltung und Wechsel zwischen den einzelnen Sensoren als Signalquelle
- AD-Wandlung und Seriellisierung der Daten mit abschließendem Senden an der Computer

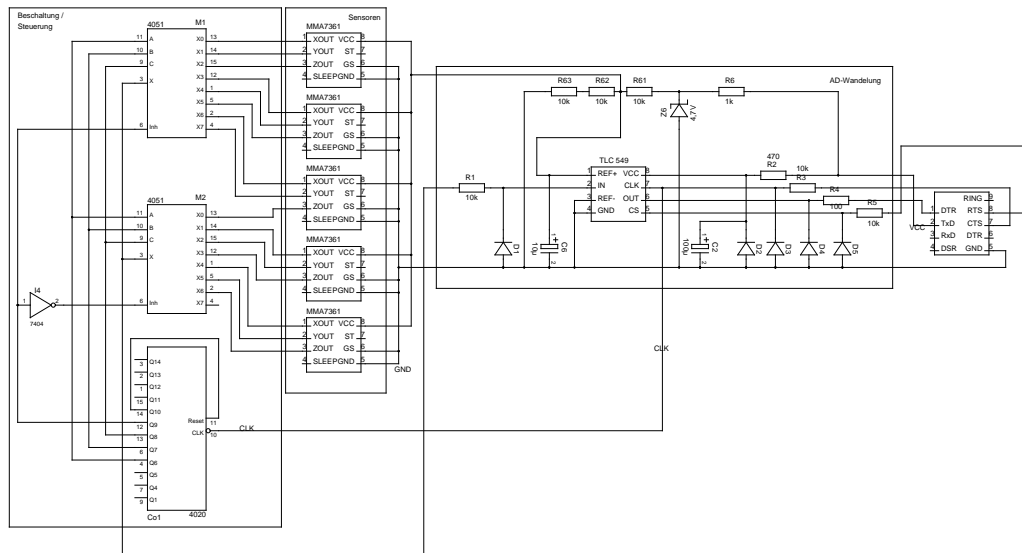


Abbildung 3.1: Gesamtschaltplan

Kapitel 4

Einzelne Teile der Hardware

4.1 Aufnahme der Messdaten

Die Bewegung der Fingerkuppen wird über fünf Beschleunigungssensoren des Typs *MMA7361LC* von Freescale Semiconductor aufgenommen. Dies sind kapazitive Beschleunigungssensoren mit einer Sensibilitätsspannweite bis 1.5g. Bei den genutzten Einstellungen beträgt die Sensitivität 800mV / g. Somit ist dieser Sensor nicht zuletzt aufgrund seines geringen Preises für die erste Testphase des Handschuhprojektes gut geeignet. Die Sensoren sind an **GND** und an eine spannungsführende Linie mit 3,13V angeschlossen. *g-select* wird ebenfalls an **GND** angeschlossen, um die Sensitivitätsgrenze von 1.5g zu wählen. Die Sensoren geben drei analoge Spannungssignale aus - eins für jede Achse. Die Höhe der Spannung gibt den aktuelle Beschleunigungswert an. Die insgesamt 15 Datenleitungen werden in den nächsten Abschnitt geführt.

4.2 Beschaltung und Wechsel

Durch die Wahl der Sensoren treffen 15 Datenleitungen gleichzeitig ein und sollen ausgelesen werden. Hier gibt es nun zwei Alternativen:

- (1) Die Daten werden serialisiert und nacheinander an den auswertenden Computer gesendet
- (2) Die Daten werden parallel an den Computer gesendet

Die Nachteile der beiden Alternativen sind klar. Bei der seriellen Datensendung gehen bei geringen Übertragungsraten viele Daten verloren. Die parallele Alternative ist zwar schneller, hat aber auch ihren Preis. Statt einem Eingang und einem AD-Wandler werden gleich 15 davon benötigt. Zu lösen ist dieses Problem über teure Laborkarten, die eigens für wissenschaftliche Zwecke hergestellt wurden. In diesem Projekt wird jedoch die erste Möglichkeit - serielle Datensendung - verwendet. Deshalb wird mit einem 14-Ripple-Counter *74HC 4020*, zwei analogen (De)Multiplexern *74HC 4051* und einem Inverter gearbeitet. Das Prinzip ist folgendes: Der Counter erhält vom Computer ein Tick-Signal und erhöht den internen Speicher um eins. Daraus folgt eine Freischaltung eines bestimmten damit verknüpften Kanals der beiden Multiplexer. Der höchste benutzte Counterausgang steuert nun, welcher der beiden Multiplexer angesteuert wird. Hat der Counter alle Kanäle der Multiplexer einmal durchgeschaltet, setzt er sich selbst auf Reset und das Intervall beginnt von vorne. Insgesamt werden somit 16 Kanäle angesteuert - von denen 15 belegt sind - und der Spannungswert jeweils an die Wandlung gesandt. Der Grund für das Auslassen der ersten drei Counterausgänge liegt in der Gesamtzeitsteuerung. In der Zeit, in der der Counter diese $2^3 = 8$ Schritte durchläuft wird das digitalisierte Signal an den Computer gesendet (siehe (Tabelle 4.1)). Multiplexer, Inverter und Counter sind jeweils an **GND** und **PWR** angeschlossen.

Counter	Offener Kanal des Multiplexers	Angezeigtes Bit des AD-Wandlers
0	0	8 (von Ch.16)
1	0	7 (von Ch.16)
2	0	6 (von Ch.16)
3	0	5 (von Ch.16)
4	0	4 (von Ch.16)
5	0	3 (von Ch.16)
6	0	2 (von Ch.16)
7	0	1 (von Ch.16)
CS-Select ausführen		
8	1	8 (von Ch.1)
9	1	7 (von Ch.1)
10	1	6 (von Ch.1)
11	1	5 (von Ch.1)
12	1	4 (von Ch.1)
13	1	3 (von Ch.1)
14	1	2 (von Ch.1)
15	1	1 (von Ch.1)
...		
65	9	8 (ab hier wird der zweite Multiplexer aktiviert)

Tabelle 4.1: Zeitlicher Verlauf der einzelnen Steuersignale

4.3 AD-Wandler

Durch die Beschaltung werden die 15 (eigentlich 16) Spannungswerte an den AD-Wandler gesendet. Dort werden sie digitalisiert auf 8 Bit und über die Datenleitung nacheinander an die serielle Schnittstelle gesendet. Der verwendete AD-Wandler *TLC548C* funktioniert nach der *Sample-and-hold*-Technik. Dabei wird vom Computer ein Signal (**CS**) zum Speichern des Messwertes gegeben. $17\mu\text{s}$ danach steht der digitale Wert am Ausgang bereit und kann durch Setzen und Löschen des Clockeingangs abgerufen werden. Auf diese Weise ist er schon serialisiert und wird in diesem Rohformat über die serielle Datenleitung gesendet. Wohlgedacht handelt es sich dabei nicht um Datenpakete nach einem bestimmten Standard der seriellen Datenleitung (siehe (Tabelle 4.2)).

ca 17 μ s	ca 2 s	ca 2 s	ca 2 s	ca 2 s	ca 2 s	ca 2 s	ca 2 s
ca 2 s							

Tabelle 4.2: Zeitlicher Verlauf der Daten auf der Datenleitung der seriellen Verbindung

Teil III

Software

Kapitel 5

Übersicht

Kapitel 6

Datenaufnahme

6.1 Serielle Schnittstellen unter Linux

6.2 Datensammlungsmethode

Kapitel 7

Neuronale Netze

7.1 Einführung in Neuronale Netze

7.2 Verwendetes Neuronales Netz

7.3 Umsetzung

Teil IV

Erfolge und Probleme

Teil V

Abschluss

Quellen

Danksagung