

**Sachbericht (Schlußbericht)  
zum FuE-Vorhaben**

Reg.-Nr.: VF120029  
FuE-Einrichtung: GFal e.V.  
Titel: WAVCAM (Ultraschall 3d-Bildsensorik)  
Projektlaufzeit: 01.01.2013 – 31.12.2014

Berlin, den 22.6.2015

Dr. Gerd Heinz, Tel. 030 814563 – 490

---

Name und Telefonnummer des Projektleiters

Firmenstempel

---

Unterschrift des Projektleiters

Rechtsverbindliche Unterschrift

# 1 Technisch-technologische Zielstellung des Vorhabens

In den letzten Jahren verbreiten sich mehr und mehr sogenannte „Natural User Interfaces“ (NUI). Smartphones mit Touchscreen werden mit Gesten statt mit Maus und Tastatur gesteuert. Im öffentlichen Raum, wie als Bedieninterface für Maschinen und Anlagen sind Touch-Displays allerdings nicht unproblematisch, verschmierte Displays werden abgelehnt. Nur berührungsfreie (touchless) Displays können einen Ausweg bilden.

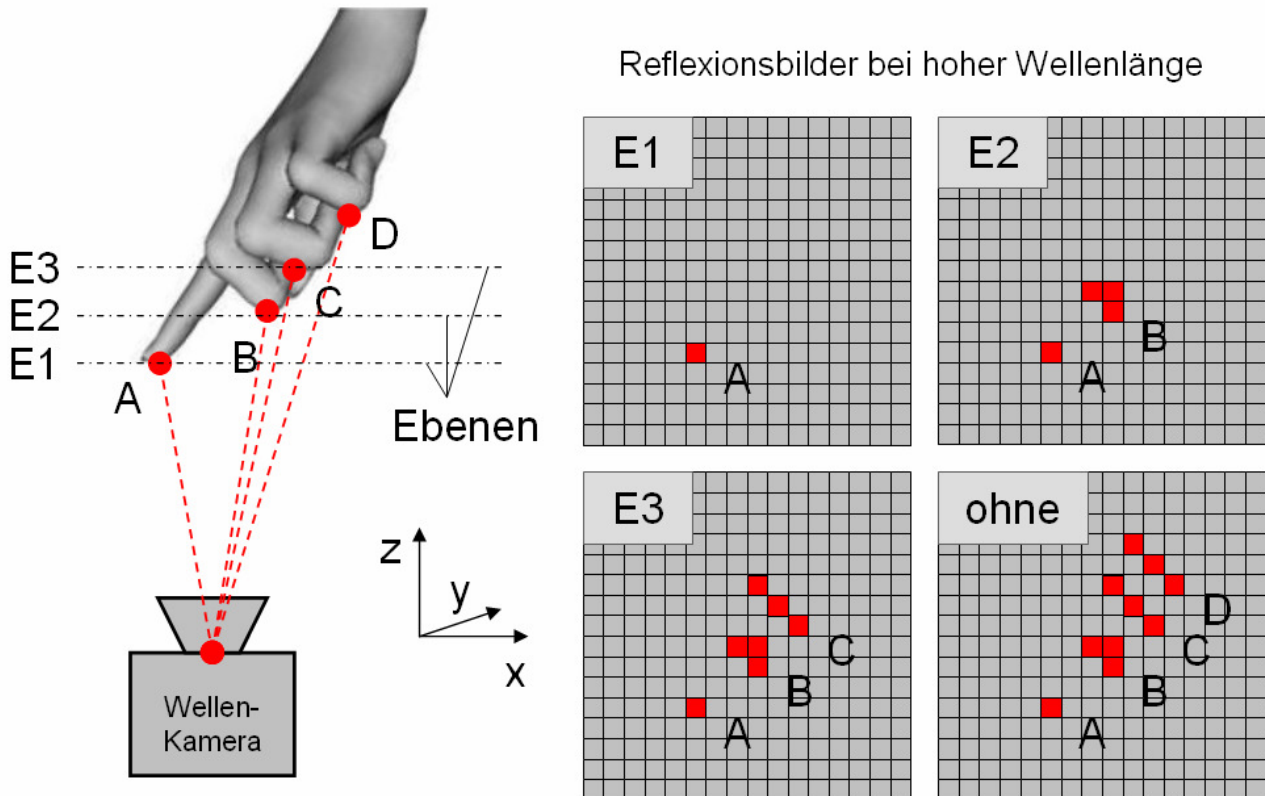


Abb. 1: Binärer Ansatz zur Informationsreduktion mit Ultraschall. Bei geeigneter Wellenlänge (um 10 mm) erscheinen nur wenige, orthogonal reflektierende Punkte im Bild. Interessiert man sich nur für das kürzeste Echo bis zur Ebene E1, bleibt allein eine Reflektionsstelle A übrig, ein virtueller Fingerdruck durch eine Ebene E1 hindurch kann erkannt werden.

Derzeit beherrscht bei berührungslosen Sensoren eine Fülle kamerabasierter Ansätze den Markt. Zur Abtastung des dreidimensionalen Raumes vor der Kamera finden Stereo-Methoden, Triangulations- und Licht-Laufzeit basierte Ansätze (Microsoft Kinect-Konsole [11]) („Time of Flight Sensor“ TOF [5]) Verwendung. Doch die Optik jener Kamera verschmutzt irgendwann, sie ist nicht wartungsfrei. Die Einführung optischer Methoden ist deshalb umstritten und kommt im öffentlichen Raum relativ schwer voran.

Fragen wir nach Wartungsfreiheit, so fallen Ultraschalldetektoren auf. An jeder Stoßstange moderner PKW sind sie zu finden. So existieren voll gekapselte Ultraschallsensoren, die praktisch lebenslang wartungsfrei arbeiten [7] können. Allerdings sind marktübliche Systeme auf einfache Laufzeitmessung begrenzt. Sie können noch kein „Bild“ sehen.

Da jetzt „embedded“ Prozessoren („all inclusive“ Prozessoren) mit erstaunlicher Leistungsfähigkeit [6] zur Verfügung stehen, sollte im Projekt ein ultraschallbasierter, bildgebender Ansatz zur Ermittlung eines dreidimensionalen Bildes untersucht und vorangetrieben werden. Die Forschungsrichtung ist insbesondere für (meist klein- und mittelständische) Firmen interessant, die sich für neuartige Maschinenbedienungen interessieren.

Im Projekt waren erste Werkzeuge und Verfahren für berührungslose, 3-dimensional arbeitende Bildsensoren basierend auf Ultraschall zu entwickeln. Mit einem interferenziellen Ansatz waren theoretische und praktische Möglichkeiten zur mikroelektronischen Realisierung als eine Art von Kamera („Wellenkamera“) für Bild- und Volumenabtastung zu untersuchen. Über eine im Verhältnis zur Rauheit *hohe Wellenlänge* sollte dabei eine *spezifische Informationsreduktion* erzielt werden. Nur die exakt orthogonal zur Wellenfront liegende Krümmung der Reflektionsflächen wirkt hier sichtbar reflektierend.

Applikationen liegen in der Prozeß-Automation, Personenzählung, als „touchless keyboard“ für Tablet- PCs, das Verfahren ist anwendbar in der Chirurgie, als Eingabegerät im öffentlichen Bereich, für Baumaschinen, für Behinderte oder als 3d- Eingabegerät.

Zusammen mit der Iris- GmbH Berlin soll eine erste Anwendung initiiert und untersucht werden als Eingabemedium für Personenzählung in öffentlichen Verkehrsmitteln.

Grundlage der Arbeiten sind Erfahrungen, die bei der Entwicklung der akustischen Photo- und Kinematographie (Akustische Kamera) gewonnen wurden.

## 2 Darstellung der erzielten Vorhabensergebnisse

Im Antrag war beabsichtigt, ein spezielles, relativ breitbandiges, analoges Ultraschallmikrofon SPM04UD5 von Knowles [8] einzusetzen, um codebasierte „Chirps“ für die Abtastung nutzen zu können. Leider war dieses bei Arbeitsbeginn nicht mehr lieferbar. Anfragen bei allen Distributoren von Knowles ergaben einen Lagerbestand von Null. Ein vergleichbares Mikrofon existierte nicht.

Als Alternative kam ein digitales MEMS- Mikrofon zum Einsatz [13], dessen Applikationsnoten den Eindruck vermitteln, auch dieses Mikrofon würde ultraschalltauglich sein. So liegt dessen Resonanzfrequenz in Abhängigkeit von der Größe des Eintrittsloches zwischen 25 und 44 kHz.

Diese Umorientierung bedeutete zweierlei. Erstens konnte damit der Hardwareaufwand extrem minimiert werden. Vorverstärker entfallen, nur noch Mikrofon und Prozessor sind nötig.

Zweitens aber – und das stellte sich erst ganz allmählich heraus – liegen kaum Erfahrungen mit digitalen Mikrofonen vor. Smartphone- Firmen behalten ihr Wissen für sich, Web-Suchen ergeben zwar tausende Treffer, die aber zeigen nur klassische Filtertheorie. Wissen zur Dezimierung, zu dynamischem Offset, zu hochfrequentem Rauschen, zur Signalfilterung oder Wissen zum Verhalten digitaler Mikrofone im Bereich zwischen 20 kHz und 40 kHz ist im WWW kaum auffindbar. Das Gebiet ist recht neu.

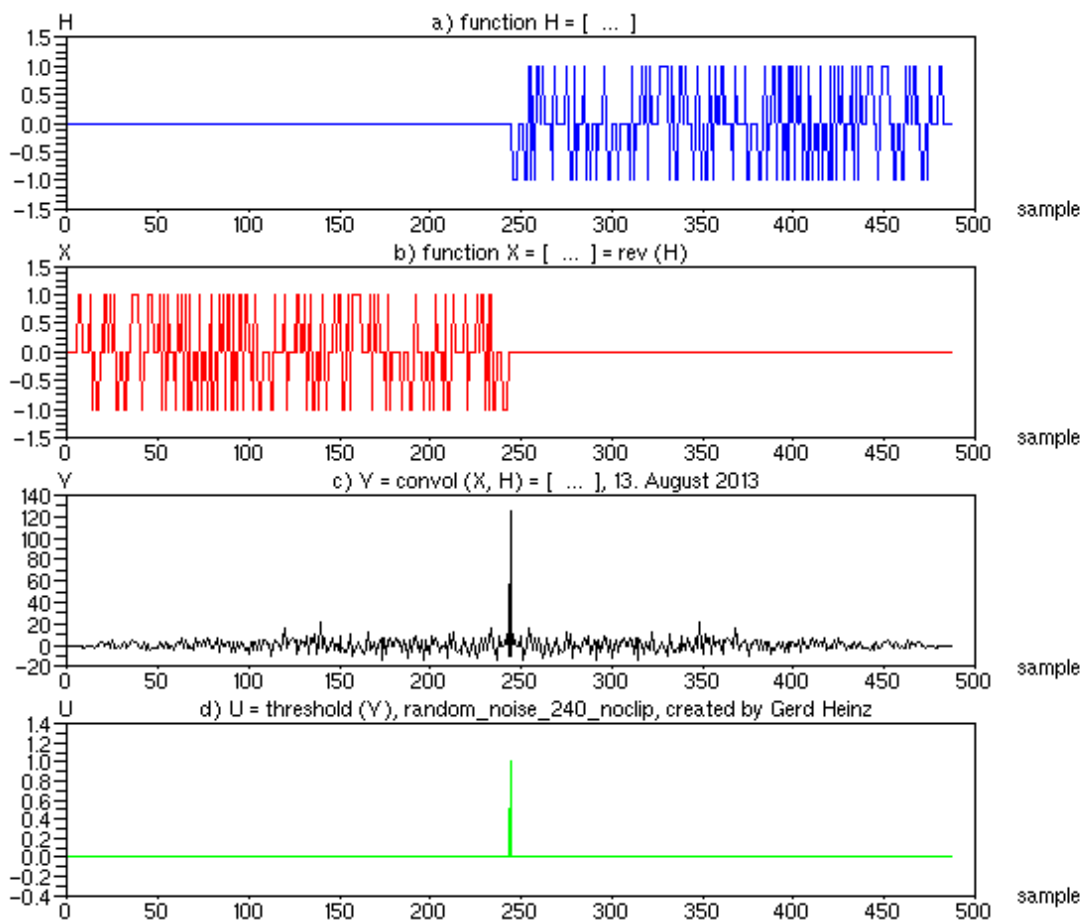
### **Signaltheorie**

Im ersten Arbeitspaket wurden Piezo- Aktoren und Piezo- Sensoren untersucht. Die Versuche zeigten, daß deren Resonanzverhalten (langsames An- und Abschwingen), wie deren extreme Richtcharakteristik für vorliegenden Zweck nicht optimal geeignet ist.

Die Nutzung von Sensoren und Aktoren mit kugelförmiger Richtcharakteristik erschien verheißungsvoller. So fiel die Wahl auf breitbandige Lautsprecher und Mikrofone. Da vergleichsweise kleine Schalleistungen zur Verfügung stehen, und die Empfindlichkeit von Mikrofonen begrenzt ist, wurden Untersuchungen zur Interferenz langer, energiereicher Zeitfunktionen vorgenommen.

Die Auslegung von im weiteren Sinne wellenbasierten Abstandsmesssystemen stellt die Frage nach maximierter Signalenergie. Einzelimpulse sind energetisch zu schwach, kontinuierliche Signale (Continuous Wave – CW) bringen technische Schwierigkeiten bei der Auswertung mit sich. Frequenz- oder phasenmodulierte Signale, wie z.B. Chirps bieten einerseits einen hohen Energietransfer, andererseits gestatten sie die exakte Ermittlung der Signallaufzeit, siehe z.B. eine hervorragende Einführung zum Thema in [10]. Unter Beachtung verfügbarer Rechenleistungen und Hauptspeichergrößen erscheinen diese Methoden nicht als zielführend.

Untersuchungen galten der Gestaltung optimal angepasster Signalverarbeitung. Ausgangspunkt war die Fragestellung, inwiefern „unkomplizierte“ Zeitfunktionen (z.B. mit Werten -1, 0, +1) benutzt werden können, die vergleichbar zu CW- Verfahren gegenüber einem Dirac- Puls mehr Signalenergie transportieren können, und die zugleich mit geringstem Soft- und Hardwareaufwand dekodierbar (faltbar) sind. Auch soll deren Faltung mit sich selbst (Schlüssel- Schloß- Prinzip) einen scharfen Spike ohne Artefakte (Nebenmaxima) liefern.



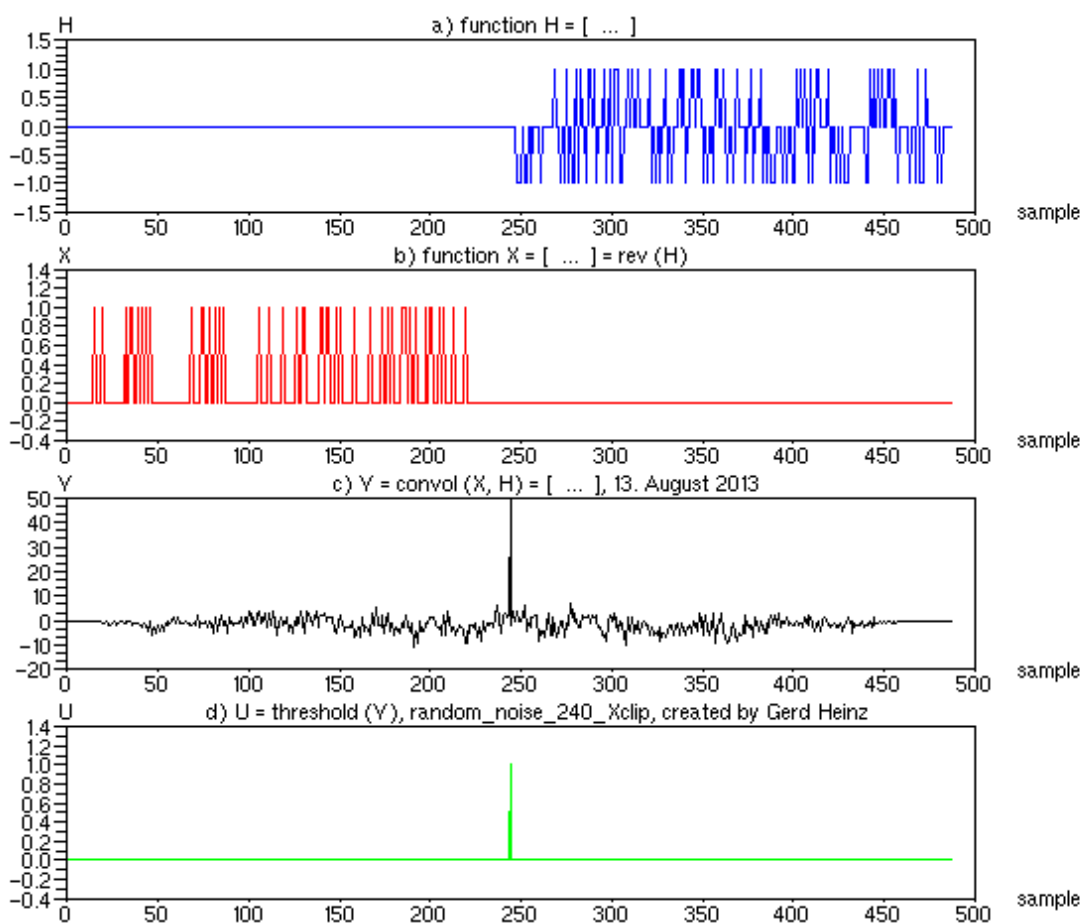
**Abb. 2:** Die Faltung einer unipolaren Rauschfunktion a) als Schlüssel mit einer unipolarer Schloßfunktion b) ergibt hohe Spikes c), d), die hervorragend zur Abstandsmessung geeignet sind, siehe [1], [2], [3].

Es entstanden Scilab- Simulationsprogramme zur Untersuchung der Eigenschaften rauschartiger, Zeitfunktionen und deren Faltungen. Zeitfunktionen und deren sphärische Realisierungen im Netzwerkmodell (Interferenznetz) wurden untersucht. Zeitfunktionen und Parameter wurden diskutiert, siehe auch AP2. Zur Realisierung geräuschfreier Sendesignale wurden insbesondere bandbegrenzte Rauschsignale untersucht, bei denen die Übertragungsfunktion H bandbegrenzt und arm an Anteilen

unterhalb von 20 kHz ist. Neben der Untersuchung von Rauschsignalen wurden das Spektrum und die Faltung von Barker- Codes experimentell untersucht.

Gegenüber Rauschsignalen haben diese zwar eine höhere Code-Effizienz. Da sie aber stets mit DC-Offset daherkommen, sind sie nur mit zusätzlicher Biphase- Modulation oder mit Phase-Shift Keying nutzbar. Auch ist deren Spektralverteilung prinzipiell ungünstiger (farbiger) als die von weißem Rauschen.

Ein hervorragendes Ergebnis dieser Untersuchung ist der Nachweis, daß bipolares Rauschen (-1, 0, +1) für Schall- oder GPS- oder Radarapplikationen bestens geeignet ist, Abb.2. Einerseits weist weißes Rauschen die flachste Spektralverteilung auf, andererseits können bipolare Rauschfunktionen beliebig lang sein, ohne daß die Interferenzqualität auch nur im mindesten leidet, siehe Abb. 2 und 3. Andererseits stehen Schaltkreise (große FPGA) zur Verfügung, die auch die Faltung langer Rauschfolgen in Echtzeit gestatten. Einzige Voraussetzung ist die, daß der Schlüssel ins Schloß paßt, d.h. daß der Schlüssel die gefaltete (zeitinvertierte) Zeitfunktion des Schloßes besitzen muß.



**Abb. 3:** Die Faltung einer unipolaren Rauschfunktion a) als Schlüssel mit deren bipolarer Schloßfunktion b) ergibt Spikes c), d), die ebenfalls hervorragend zur Abstandsmessung geeignet sind, siehe [1], [2], [3]. Besonderer Vorteil dieser Variante ist die rein binäre Realisierbarkeit des Schloßes.

Auch die Kombination zwischen unipolarer (0, 1) Wellenfunktion (z.B. nervlicher Spike) und bipolarem Detektor (Synapse) ist produktiv, wenngleich sie auch nur den halben Spitzenwert liefert. Besonderer Vorteil dieser Variante ist die Tatsache, daß zur Decodierung ein reines Binärsignal genutzt werden kann.

Damit kann die Frage, ob unser Nervensystem als Interferenznetzwerk funktioniert, endgültig bejaht werden. Im Nervensystem laufen monopolare Signale (Spikes) in bipolare Synapsen der Nervenzellen ein. Der systemtheoretische Horizont dieser Kombination war bislang unklar. Während die Kombination zwischen monopolarer Schloß und monopolarer Schlüssel ein dynamisches Offset liefert, bleibt eine gemischte Kombination nach Abb.3 nahezu offsetfrei.

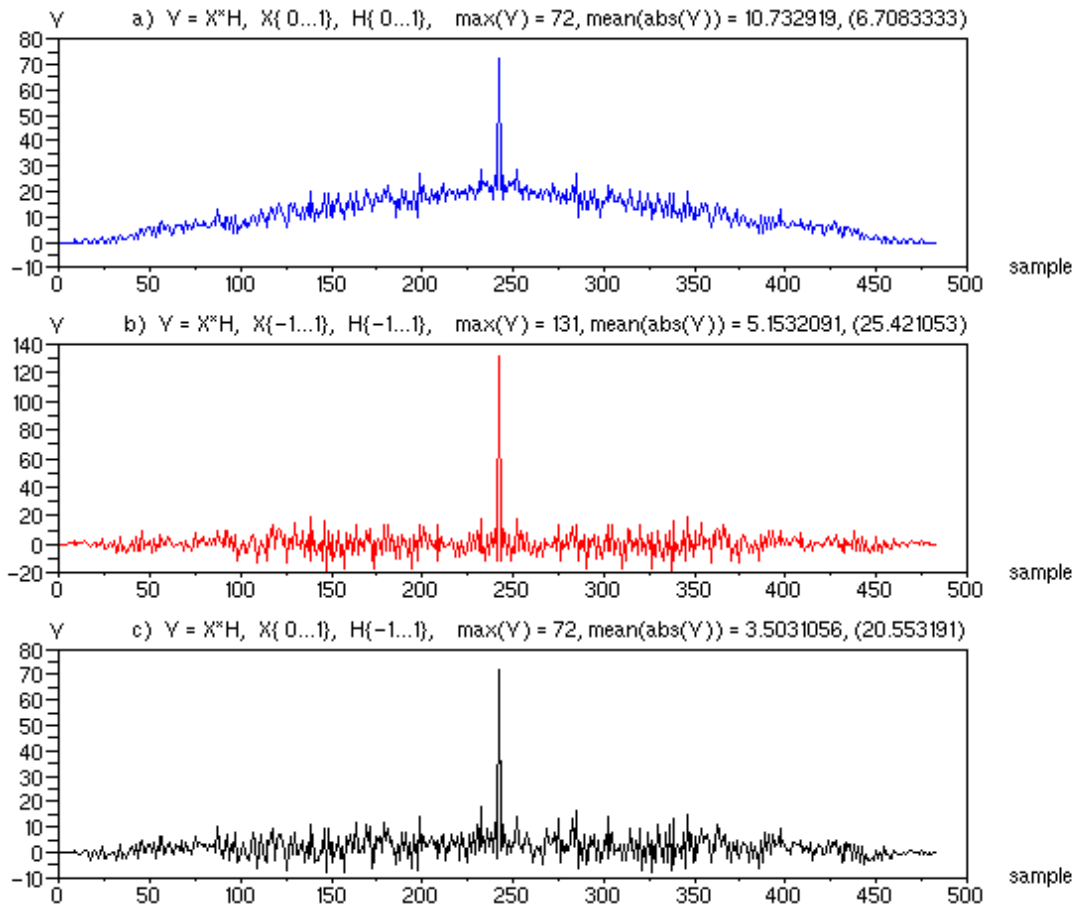


Abb. 4: Interferenzqualität vergleichbarer Kombinationen aus Schlüssel X und Schloß H: a) monopolar/monopolar, b) bipolar/bipolar, c) monopolar/bipolar; Quelle [3]. Man erkennt, daß Kombination a) für eine unkomplizierte Schwellwertbildung ungeeignet ist. Kombination b) stellt das Optimum dar. c) liefert im Vergleich mit b) nur einen halb so hohen Interferenzwert, ist aber bestens detektierbar.

Eine Untersuchung der Bandbegrenzung (Hochpaß) des Rauschens führte immer zu einer schlechteren Peak- Abhebung. Untersuchungen dazu zeigten allerdings noch keine Lösung.

Um den Zusammenhang zwischen Zeitfunktion als Schlüssel und Filterstruktur (FIR) als Schloß herzustellen, wurden Transformationsregeln erstellt und in Scilab programmiert [3]. Ist eine FIR-Filterstruktur (T,W,fs) als Schloß gegeben, kann daraus der Schlüssel, eine Zeitfunktion oder Transferfunktion H bestimmt werden.

Es bedeuten:

- T: Vektor der Verzögerungszeiten,
- W: Vektor der Gewichte und
- fs Abtastfrequenz.

```

function [H] = trans(T,W,fs);
if length(T) == length(W) then
    T = T * fs;           // apply sample rate of H
    T = round(T);        // T becomes index - integer
    H = 1:max(T); H = H * 0; // create an empty H
    for i = 1:length(T), // for all T(i), W(i)
        j = T(i),       // delay becomes the H-index
        H(j) = H(j) + W(i), // add the weight to H
    end // for
else // if
    printf('\n\nerror: T and W have different size\n');
end // if
endfunction;

```

Umgekehrt, ist eine Zeitfunktion H mit deren Samplingfrequenz fs gegeben, kann daraus das passende Schloß als FIR-Filter bestimmt werden.

```

function [T,W] = net(H,fs); // returns T and W
    j=1;
    for i=1:length(H) // H index i
        if H(i) == 0 then ; // do nothing
        else // write the value to W, the index to T
            W(j) = H(i);
            T(j) = i;
            j = j+1;
        end; // if
    end; // for
    T = T ./ fs; // remove sample rate
    T = T - min(T); // scaling
endfunction;

```

Um ein Beispiel zu geben: Es sei eine FIR-Filterstruktur mit T und W gegeben,

$$T = [\tau_1, \tau_2, \tau_3] = [5, 3, 8], \quad W = [w_1, w_2, w_3] = [1, 0.5, 1].$$

Zu dieser korrespondiert die reduzierte (Index r) Transferfunktion T und Gewichtsfunktion W

$$T_R = [\tau_{R1}, \tau_{R2}, \tau_{R3}] = [0, 2, 5], \quad W_R = [w_{R1}, w_{R2}, w_{R3}] = [.5, 1, 1].$$

Scilab liefert einen Beschreibungsfile \*.txt, dessen Inhalt wie folgt aussieht:

```

sample rate      fs = 1 kHz
original delays  T = [5 3 8]
original weights W = [1 0.5 1]
reduced delays   Tr = [0 2 5]
reduced weights  Wr = [0.5 1 1]
transfer fct H = [0 0 0 0.5 0 1 0 0 1 0]
key input       X = [0 1 0 0 1 0 0.5 0 0 0]
output Y = [0 0 0 0 0 1 0 1 1 0 2 0 0 1 0 1 0 0 0]
thresh U = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
abs(F) = [2.5 1.4 0.5 1.1 1.8 1.9 0.5 1.9 1.8 1.1 0.5 1.4]
generator: transfuncnet.sce, 13-Aug-2013
www.gfai.de/~heinz/techdocs

```

Die Scilab-Programme `transfuncnet.sce` und `noiseconv.sce`, welche Abb. 2 bis 4 und den Textfile erstellen, können im Internet [2], [4] heruntergeladen werden. Die Präsentation und Diskussion dieser ersten Untersuchungsergebnisse erzielte Aufmerksamkeit auf der wissenschaftlichen Konferenz [2].

## Hardwareentwicklung

Wie bemerkt, führte die Verfügbarkeit eines digitalen, bedingt ultraschalltauglichen Mikrofons [13] zur Änderung der Hard- und Softwarestrategie. Die Mikrofone liefern ein pulsdichtemoduliertes Signal der Form Abb.5. Um hinreichend hohe Rechengeschwindigkeit zu erhalten, kommt der derzeit kompakteste und schnellste embedded Prozessor [6] zum Einsatz.

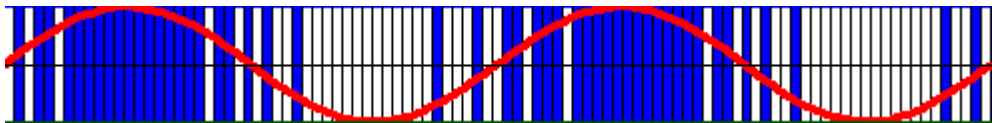


Abb. 5: Pulsdichtemodulation. Verwendete Mikrofone liefern je einen Ein-Bit Datenstrom mit zwei Megahertz. Quelle: Wikipedia.

Damit entfiel ein gewisser Teil der für Kleingeräte und Material vorgesehenen Ausgaben. Der Kern der Schaltungstechnik besteht aus Prozessor, Mikrofonen und Lautsprecher, Abb.6. Mikrofon-Vorverstärker entfallen. Nur Treiber- IC für Takt (CLOCK) und Lautsprecher (SPEAKER) sind erforderlich.

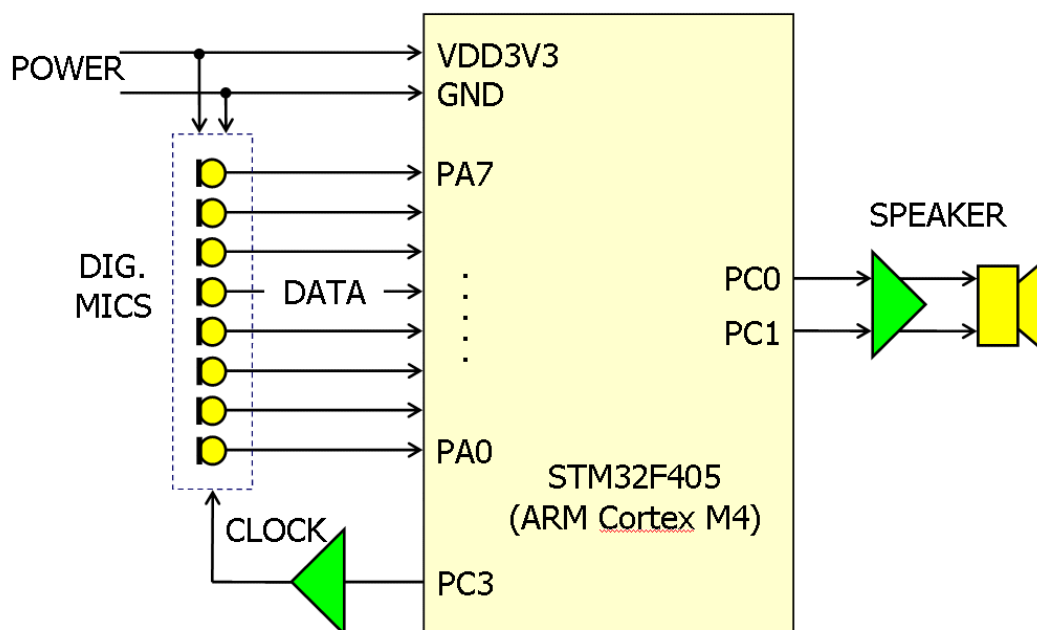


Abb. 6: Mikrocontroller-Interface der Mikrofone. Sie erhalten Betriebsspannung sowie ein verstärktes Taktsignal CLK.

Mit dieser Auslegung wird eine extreme Miniaturisierung und Verringerung des Hardwareaufwandes möglich. Zur Grundstruktur hinzu kommen allerdings noch Quarz, Lautsprecher-Treiber, Transverter und Pulstransformator (alternativ bestückbar) für den Lautsprecher, Spannungsversorgungen sowie Schnittstellen-Treiber für RS485, CAN und USB, siehe Abb.7.

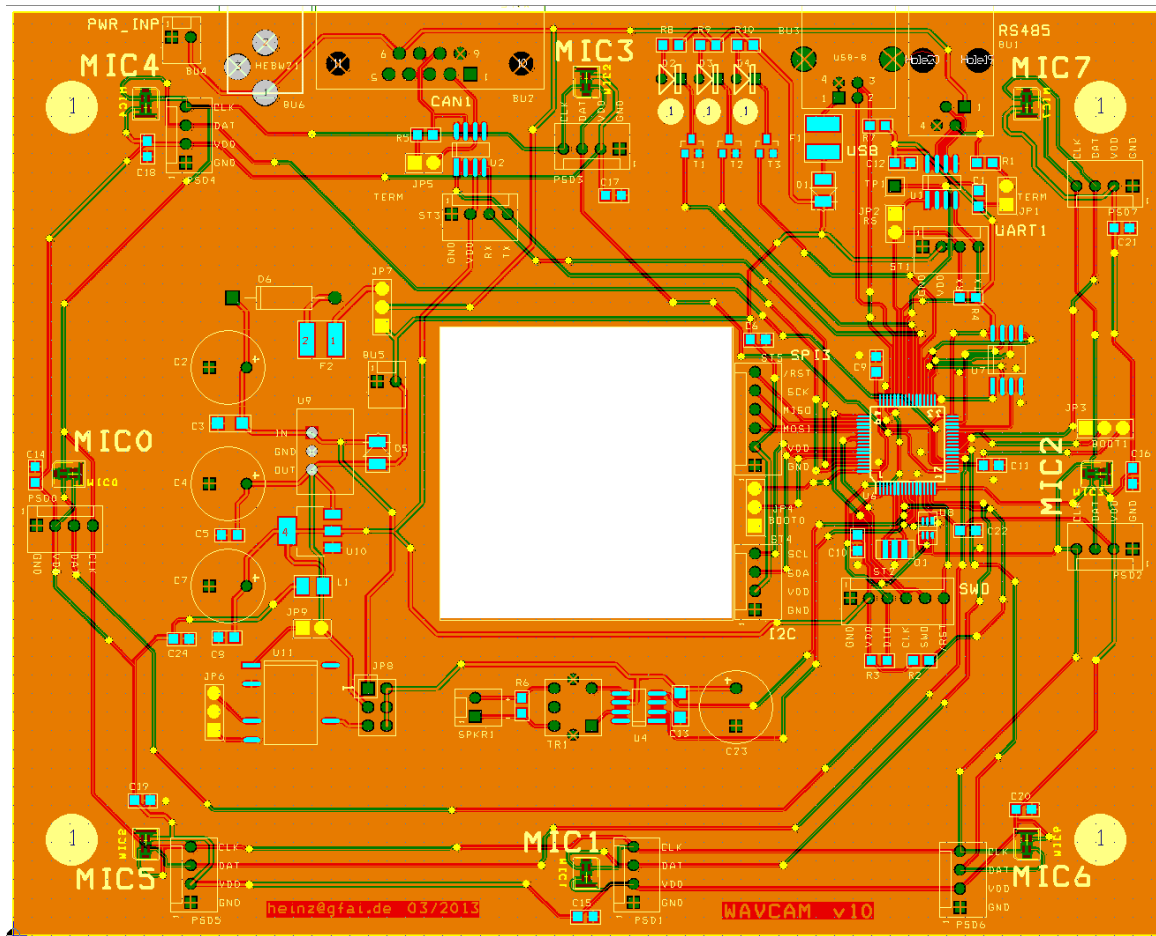


Abb. 7: Layout der entwickelten Leiterkarte, Ansicht TOP. Die Mikrofone liegen auf der Rückseite. Der Lautsprecher liegt in der Aussparung in der Mitte.

Die Betriebsspannungsversorgung wurde relativ universell gestaltet. Einerseits ist es möglich, das Gerät nur mit 5 Volt über USB oder RS485 zu speisen. Andererseits kann ein DC/DC-Wandler bestückt werden, der Betriebsspannungen von bis zu 24 Volt auf 5 Volt wandelt.

Um maximalen Freiraum bei der Wahl der Lautsprecher zu haben, kann der Lautsprecher- Treiber- IC mit 5, 15, oder 30 Volt betrieben werden. Für Piezolausprecher kann ein 1:10 hochsetzender Pulstrafo bestückt werden.

Die Anordnung der Mikrofone in Abb.11 wurde so gewählt, daß auch eine 4-Kanal Variante abgeleitet werden kann. Diese hätte die Mikrofonnummern 0 bis 3 oder 4 bis 7.

Die Konstruktion ist so ausgeführt, daß Decken- oder Wandbefestigung möglich ist, siehe Abb.12. Dabei wird die Rückseite fest am Objekt verklebt oder verschraubt. Nach Lösen der von vorn zugänglichen Schrauben kann die Frontseite komplett mit Elektronik abgenommen werden. Damit können Wartungsarbeiten unkompliziert durchgeführt werden.



Abb. 8: *Gerät in Minimalausstattung (USB) mit dynamischem Lautsprecher und direkter Speisung mit 5 Volt über USB- Buchse. Alle Bauteile befinden sich auf dieser Leiterplatte. Die Mikrophone liegen hinten.*



Abb. 9: *Gerät in maximaler Ausbaustufe mit Piezo-Lautsprecher und Pulstransformator. Zu erkennen sind Anschlüsse für CAN, USB, RS485 und Betriebsspannung (VDD 5...24V). Die Lautsprecher-Treiberspannung kann mit Transverter wahlweise 5V, 15V oder 30 Volt betragen.*

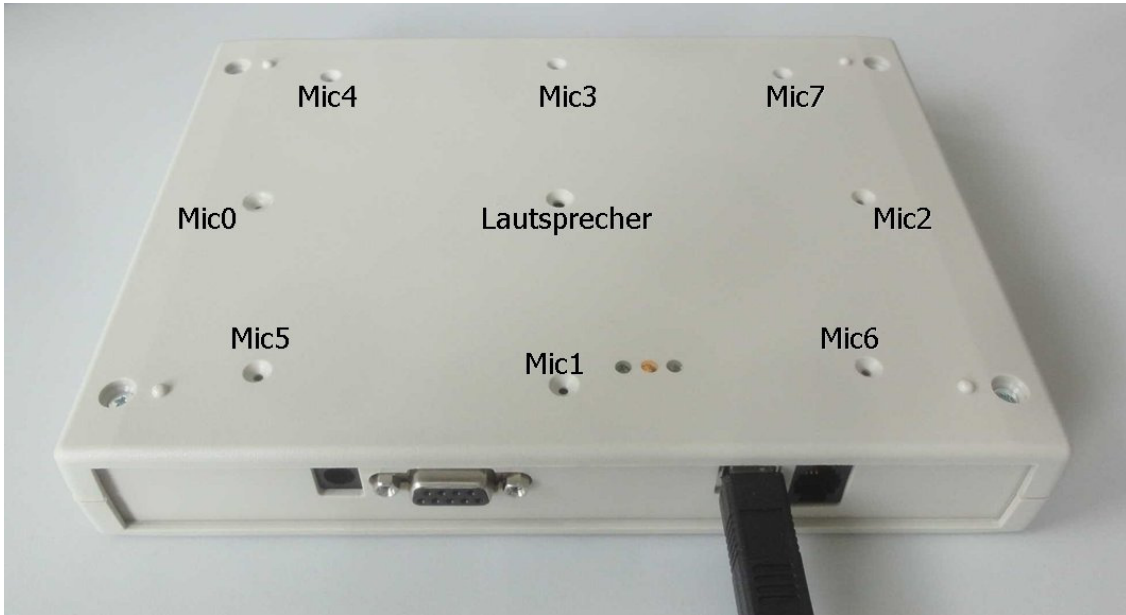


Abb. 10: Gesamtansicht. Auf der Oberseite gruppieren sich 8 Mikrophone um den zentralen Lautsprecher, zu erkennen an den trichterförmigen Schallöchern.

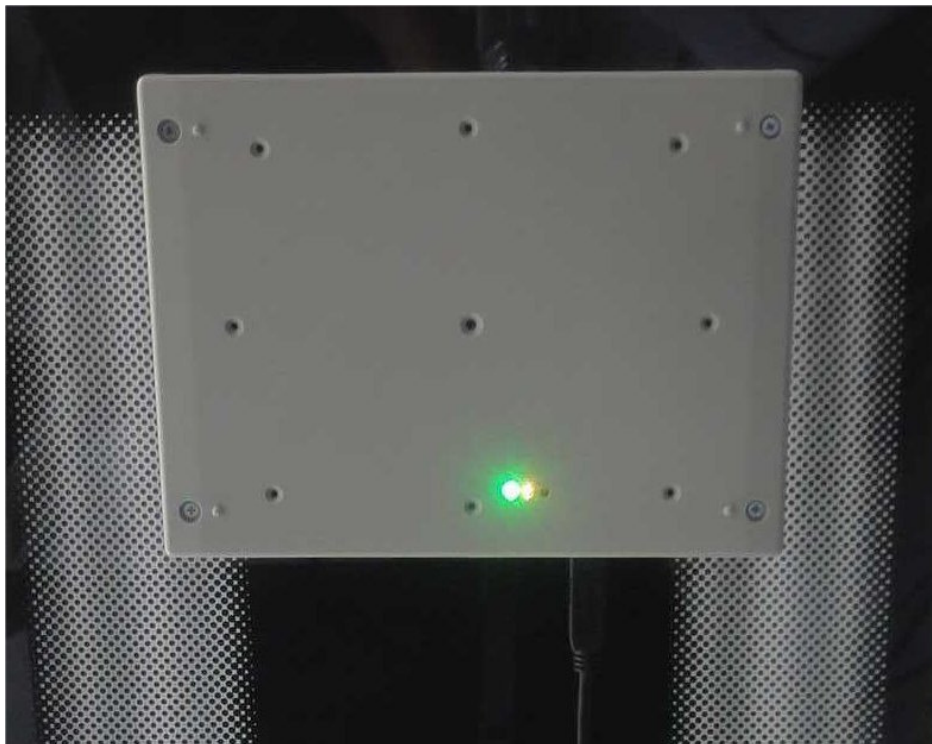


Abb. 11: Deckeninstallation. Die Konstruktion ist so ausgeführt, daß Decken- oder Wandbefestigung möglich ist.

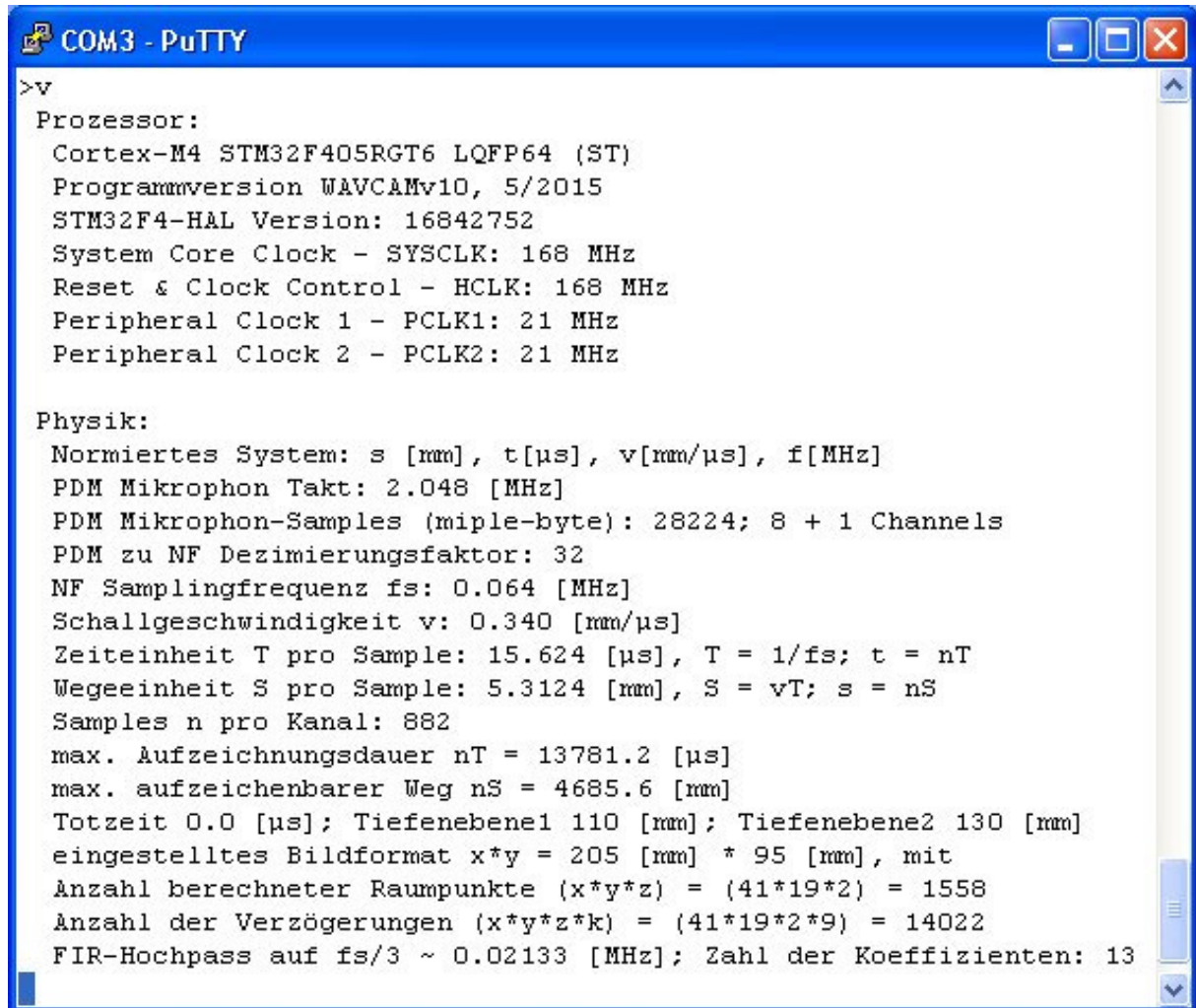


## Rekonstruktionsalgorithmen

Konform zum Antragsziel wurden drei Arbeitsmodi implementiert:

- Im Mode m „USB-HID Maus“ erzeugt die Bewegung eines Gegenstandes im Bildfeld der Kamera eine Bewegung des Mauszeigers.
- Im Mode p „periodic sampling“ werden Bilder geringer Auflösung berechnet. Hier wird eine permanente Erfassung von durch das Bild laufenden Personen möglich sein.
- Auf Anforderung mit dem Befehl i wird für Testzwecke jeweils ein Bild versandt.

Mit dem Befehl v werden interne Variablenwerte angezeigt, die zum Verständnis der Funktion beitragen können.



```
>v
Prozessor:
Cortex-M4 STM32F405RGT6 LQFP64 (ST)
Programmversion WAVCAMv10, 5/2015
STM32F4-HAL Version: 16842752
System Core Clock - SYSCLK: 168 MHz
Reset & Clock Control - HCLK: 168 MHz
Peripheral Clock 1 - PCLK1: 21 MHz
Peripheral Clock 2 - PCLK2: 21 MHz

Physik:
Normiertes System: s [mm], t[us], v[mm/us], f[MHz]
PDM Mikrophon Takt: 2.048 [MHz]
PDM Mikrophon-Samples (miple-byte): 28224; 8 + 1 Channels
PDM zu NF Dezimierungsfaktor: 32
NF Samplingfrequenz fs: 0.064 [MHz]
Schallgeschwindigkeit v: 0.340 [mm/us]
Zeiteinheit T pro Sample: 15.624 [us], T = 1/fs; t = nT
Wegeeinheit S pro Sample: 5.3124 [mm], S = vT; s = nS
Samples n pro Kanal: 882
max. Aufzeichnungsdauer nT = 13781.2 [us]
max. aufzeichnenbarer Weg nS = 4685.6 [mm]
Totzeit 0.0 [us]; Tiefenebene1 110 [mm]; Tiefenebene2 130 [mm]
eingestelltes Bildformat x*y = 205 [mm] * 95 [mm], mit
Anzahl berechneter Raumpunkte (x*y*z) = (41*19*2) = 1558
Anzahl der Verzögerungen (x*y*z*k) = (41*19*2*9) = 14022
FIR-Hochpass auf fs/3 ~ 0.02133 [MHz]; Zahl der Koeffizienten: 13
```

Abb. 13: Mit dem Befehl v werden interne Variablenwerte angezeigt.

So werden in Abb.13 gerade 28224 Mikrophonbytes aufgezeichnet. Mit einer Decimation von 32 werden daraus 882 Samples pro Kanal. Es werden 41 mal 19 mal 2 gleich 1558 Raumpunkte berechnet. Da jeder Raumpunkt eine Verzögerung zu 8 Mikrophonen und einem Lautsprecher hat, werden 14022 Verzögerungszeiten berechnet.

Ein Blick auf Abb.14 zeigt den Dezimierungsprozeß (Decimation). Aus je 32 Mikrophon- Bits, die mit 2,048 MHz einlaufen, wird für jeden Kanal ein Sample (int32) mit 64 kHz berechnet.

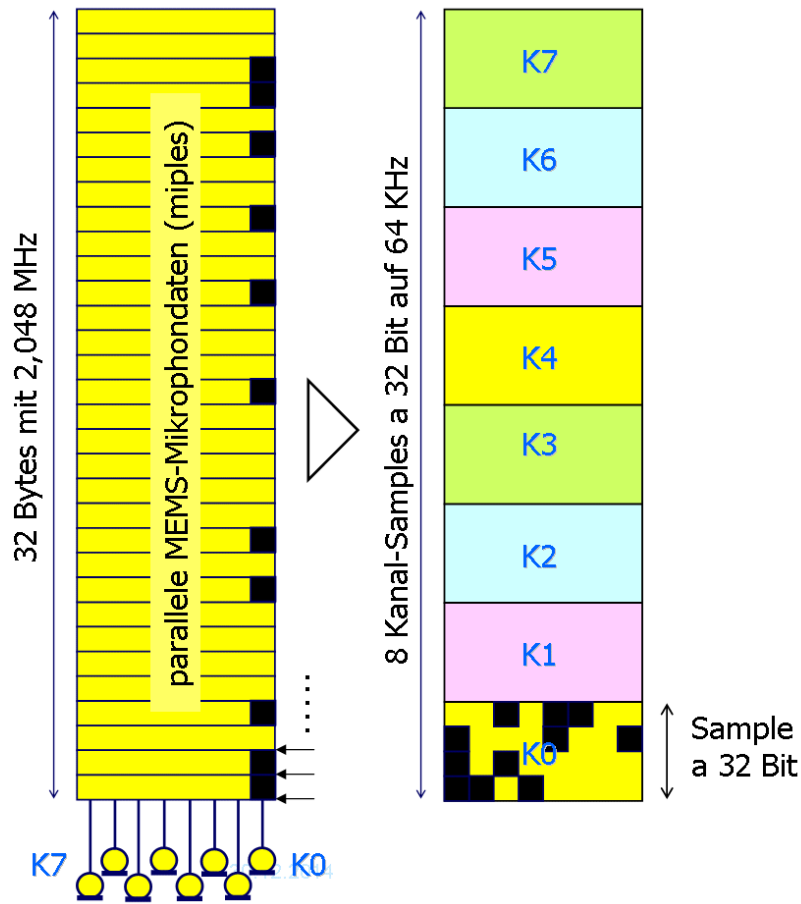


Abb. 14: Im Dezimierungsprozeß wird aus 32 Mikrofonbits (links) pro Kanal ein Sample (rechts).

Aus Gründen sparsamer Speicherverwendung werden Mikrofonbytes und Samples über eine Union im selben Speicher gehalten:

```

union b2i { // Doppelnutzung des Mic-Buffers für Miples und Samples
    uint8_t    Byt[32];    // Miples
    int32_t    Int[8];     // Samples
};    union b2i    Sample[sampanz];

// Aufruf z.B. Sample[x].Byt[y] oder Sample[x].Int[y]
// 8 Mics füllen ein Byte, PA0...PA7

```

Beim Dezimierungsprozeß sind die in je 32 Bit des Mikrophons vorkommenden Nullen und Einsen abzuzählen. Der Wert ist dem Vorgängersample zu addieren.

Vor der Dezimierung ist der Speicher mit Miples (Mikrophon-Bytes) gefüllt:

```

>a // Miples
[Anordnung der Mic-Bits im Byte: 7,6,5,4,3,2,1,0]
[0] 67 15 EA DC 33 EE 79 3 CC F7 90 6F 1B AC 96 70 EF 75 1B C8 62 95 2F
FE D1 80 7E 2F 50 EF B3 DD
[1] 8 76 A5 4B 9E F0 2F D0 6F 30 D7 29 DE F7 9 3E C4 E3 58 AE 93 7D 20 D7
7C CB 2F 90 66 B9 D7 69
[2] 16 EC 93 7E 89 76 49 A6 FC 13 ED D3 3E 8 F3 D F6 41 FE D F2 BD 21 4E
F2 9F 40 3F C0 BF 76 C9
.....

```

```

[879] C7 FA 7D 86 FB 5 38 F6 CB 4 FF 73 8C 29 76 DB D4 2B F0 D F7 9E 60
AF 53 C FE F3 D 70 FF 82
[880] AD 5A F5 4B 34 9B 44 EF 30 D7 AA 7D 87 78 E6 1B FC 9 F6 53 AD F6 B
D5 2A D4 69 E F3 3D D6 AA
[881] FD 1 7E D3 E5 1A 24 DF 20 DF EA 75 BB C F7 4F 80 F8 13 7E E5 18 A7
D6 7D 4B AA B4 5B F5 2B C6
[bytes transfered: 28224]

```

Nach der Dezimierung stehen im selben Speicherbereich die Samples (int32):

```

>c
[sampl, K0,      K1,      K2,      K3,      K4,      K5,      K6,      K7]
[0],    -5,     -6,     -9,     -4,     -11,    -8,     -10,    -4
[1],    -2,     -7,     -7,     -4,     -9,     -8,     -8,     -5
[2],    -6,     -7,     -9,     -5,     -6,     -9,     -10,    -4
[3],    -5,     -7,     -9,     -3,     -11,    -9,     -10,    -5
...
[880],  0,      0,      0,      0,      0,      0,      0,      0
[881],  0,      0,      0,      0,      0,      0,      0,      0
[samples transfered: 882]

```

Im Detail ist der Dezimierungsalgorithmus zu komplex, um hier sinnvoll dargestellt zu werden. Im Kern wird jedes Mikrophon- Bit ausmaskiert. In Abhängigkeit von dessen Wert wird ein Puffer hoch oder runter gezählt.

Aus Rechenzeitgründen wurde die Lösung mittels Rauschfunktion aus AP1 nicht eingesetzt. Wir begnügen uns mit einem sog. Integer- Bandpaß Filter (reduzierte Form). Das ist ein FIR- Filter, welches zur Vermeidung von Multiplikationen als reduzierte Form mit den Faktoren 0, +1 und -1 auskommt.

```

// Bandpass auf fs/4, 64kHz / 4 = 16 kHz, Prinzip:
uint8_t nrcoeff = 7;
const float coeff[nrcoeff] = {1, 0, -1, 0, 1, 0, -1};
uint8_t gain = nrcoeff

// Klassische FIR-Form:
sample[k][n] =
    sample[k][n] * coeff[0] +
    sample[k][n+1] * coeff[1] +
    sample[k][n+2] * coeff[2] +
    sample[k][n+3] * coeff[3] +
    sample[k][n+4] * coeff[4] +
    sample[k][n+5] * coeff[5] +
    sample[k][n+6] * coeff[6];

// reduzierte Form:
sample[k][n] =
    sample[k][n] - sample[k][n+2] +
    sample[k][n+4] - sample[k][n+6];

gain = (abs(sum(coeff))); // Verstärkung berechnen
sample[k][n] = sample[k][n] / (float) gain; // Amplitude normieren

```

Optional wird ein Integer- Bandpass für 64/3 kHz benutzt. Da die absolute Amplitude hier uninteressant ist, kommt z.B. eine Approximation nach Abb.15 in Frage.

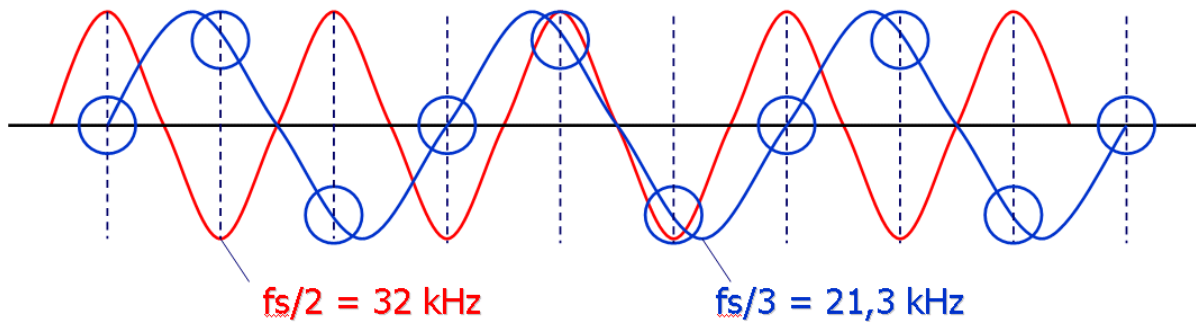


Abb. 15:  $fs/3$  Integer- Bandpaß auf 21,3 kHz. Gestrichelte Linie: Sample

Im C-Code sieht dieser Bandpaß „ohne Koeffizienten“ z.B. wie folgt aus:

```
void bandpass(void) {           // Filter auf fs/3 = 21,333 kHz
    int i, k;                   // ohne Gain-Korrektur
    for( k=0; k<8; k++ ) {     // für alle Kanäle

        // FIR-Filter
        for ( i=0; i<sampanz-nrcoeff; i++ ) { // für alle Samples
            Sample[i].Int[k] = (
                Sample[i + 0].Int[k] - Sample[i + 1].Int[k] +
                Sample[i + 3].Int[k] - Sample[i + 4].Int[k] +
                Sample[i + 6].Int[k] - Sample[i + 7].Int[k] +
                Sample[i + 9].Int[k] - Sample[i +10].Int[k] +
                Sample[i +12].Int[k] - Sample[i +13].Int[k])/2;
        } // Restliche Samples mit Nullen überschreiben:
        for ( i=sampanz-nrcoeff; i<sampanz; i++ )
            Sample[i].Int[k] = 0;
    }
}
```

Vergleichbar kann man eine Rauschfunktion als Schloß nutzen.

Zur Berechnung des Interferenzraumes werden bereits beim Setup des Prozessors die Verzögerungsmasken aller Punkte berechnet. Das spart viel Rechenzeit. Dazu werden für jeden Punkt die Laufzeiten zu den 8 Mikrofonen und zum Lautsprecher bestimmt. In einer zweiten Funktion wird die Laufzeit zum Lautsprecher zur Mikrofonlaufzeit addiert. Für jeden der z.B.  $(x, y, z) = (41 \cdot 19 \cdot 2) = 1558$  Raumpunkte (= Masken) wird ein Vektor von 8 Werten abgelegt, insgesamt also werden  $1558 \text{ mal } 8 = 12464$  normierte Verzögerungen einmalig anfangs berechnet und dann im RAM abgelegt.

Zur Berechnung des Interferenzraumes für Maussteuerung wird anders vorgegangen. Hier werden die Werte von je zwei diametralen Mikrofonen der x-Achse und der y-Achse als hyperbolische Projektion interferiert (gefaltet, kreuzkorreliert). Der gesuchte Punkt befindet sich dann im Schnittpunkt der Hyperbel- Äste. Das spart Rechenzeit.

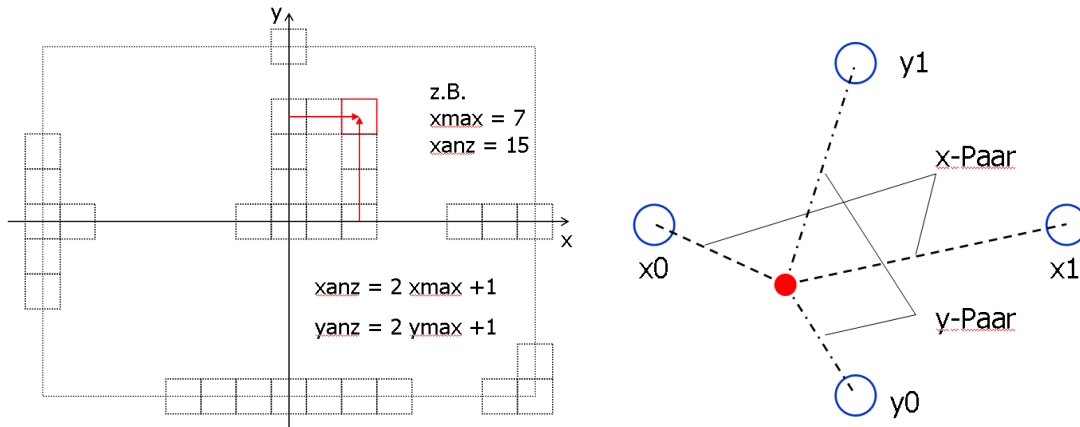


Abb. 16: Links: Voxel- Darstellung: Zur bildartigen Rekonstruktion werden symmetrische Voxelebenen (x,y) genutzt. Rechts: Bei der Maussteuerung werden zwei unabhängige, hyperbolische Faltungen je für x- und y-Achse ausgeführt, um den Ereignisort (rot) zu ermitteln.

## Softwareentwicklung

Wie schon eingangs erwähnt, wurde der zu Projektbeginn leistungsstärkste „all inklusive“ Prozessor gewählt. Es kam ein Cortex M4 von ST zum Einsatz, der mit 168/180 MHz das Feld anführte und der nach Datenblatt 196k RAM haben soll. Extern ist nur ein Quarz anzuschließen. Der Prozessor ist eine Weiterentwicklung des ARM7. Die Dokumentation umfaßt mehrere Tausend Seiten. Sie ist allerdings so lückenhaft und verwirrend, daß man beim Einstieg ohne die Hilfe von Google- Kochrezepten praktisch nicht von der Stelle kommt.

Als Entwicklungssystem kam Raisonance Ride7 zum Einsatz. Mit dem Programmer/Debugger Rlink-STD ist es ohne Probleme möglich, den Prozessor im Flash-Boot über Serial Wire Debug (SWD) zu programmieren und zu debuggen. RAM-Boot funktionierte nur sporadisch. Gegenüber dem 20-poligen JTAG- Stecker besitzt SWD neben VDD3V3 und GND minimal nur zwei Leitungen (CLK und DATA). Die Umgebung lehnt sich an Keils MDK-ARM an.

Zunächst wurde erfolglos versucht, hunderte Prozessoreinstellungen in klassischer Manier zu initialisieren. Versuche mit den Entwicklungskits STM32F4 Discovery und STM32F4 Nucleo brachten zwar sofortige Möglichkeit zu programmieren, ließen aber keine Chance erkennen, das gewünschte Setup „zu Fuß“ zu erreichen.

Erst das von ST angebotene Cube- Tool brachte eine gewisse Abhilfe. Mit Cube von ST kann relativ leicht ein Setup des Prozessors erreicht werden. Allerdings ist der Preis hoch. Cube bringt rund zwei Dutzend vollkommen ineffizient „STM32xx\_HAL\_Driver“ mit sich (\*.C und \*.H Files).

Und erst Wochen später bemerkt man, daß der superschnelle Prozessor nun für die Ausgabe eines einzigen Bits (Mikrophontakt 2,048 MHz) rund 370ns (etwa 62 Takte) braucht. Unzumutbar, wenn man zur Erzeugung eines Mikrophontaktes eine Umschaltung mit 4...7 MHz benötigt. Alternativen, wie SysTick oder Timer funktionieren ebenfalls nur bis zu einem Megahertz. Zur Nutzung des PWM-Modes der Timer stehen nur bestimmte Ausgabepins zur Verfügung, diese waren bereits anders belegt.

Eine zweckentfremdete Nutzung der SPI- oder I2S-Clocks zur Takterzeugung in der Art vom STM32 Discovery Board [14] versagte an Software- Konfigurationsproblemen. Es kamen einige unvorteilhafte Umstände zusammen.

Nach geraumer Zeit wurde zusätzlich klar, daß über GNU-C in Ride7 Umgebung nur 64k RAM von 192k erreichbar sind. Da die Berechnung eines Interferenzraumes sehr speicherintensiv ist, wurde

das Systemkonzept auf Mehrfachnutzung des RAMs umgestellt. Fortan werden bei der Decimation die Mikrofondaten (Miple-Bytes, 2 MHz) von den Samples (int32, 64kHz) überschrieben.

Die Probleme führten dazu, daß zum Aufsetzen der Entwicklungsumgebung sehr viel mehr Zeit benötigt wurde, als in den Arbeitspaketen vorgesehen war. Es wird klar, auf welche Risiken sich kmU beim Start mit einer ARM- Prozessorlinie einlassen.

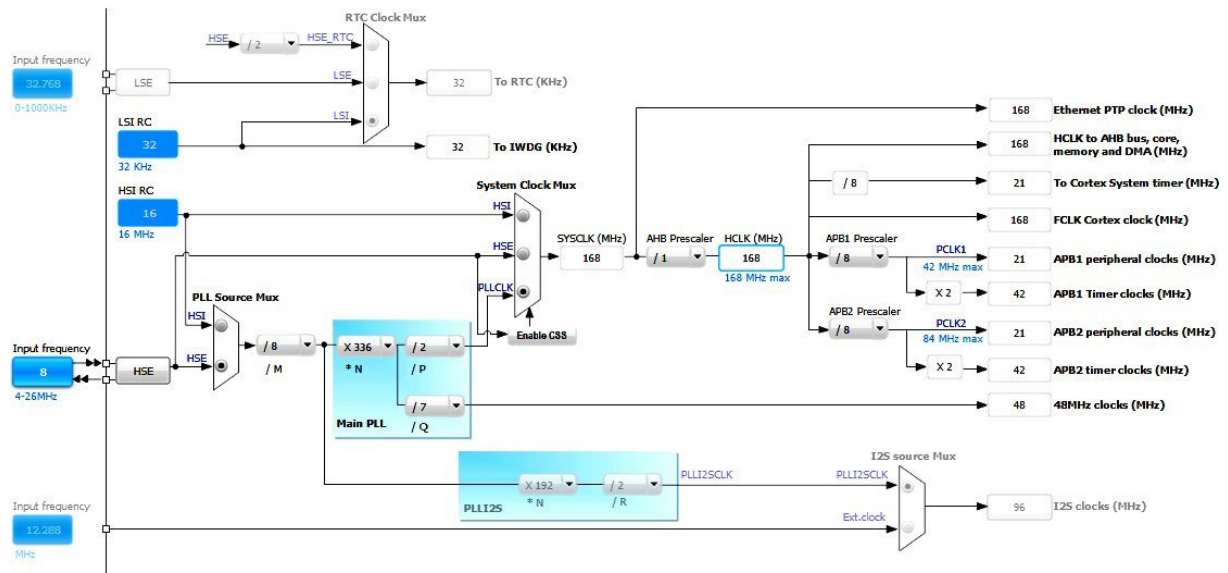


Abb. 17: Mit dem HAL- Entwicklungssystem Cube von ST wurde ein Setup des Taktsystems, des Pinsystems, der peripheren Module, des Kerns, der peripheren Schnittstellen, des Interruptsystems und des Prozessors erreicht.

Aufgrund der Speicherplatzprobleme konnte eine Decimation-Library (pdm\_filter.h unter utilities\stm32f4\_discovery [14]) nicht benutzt werden. Eine spezifische Eigenentwicklung war erforderlich.

Im Detail wurde die Softwareentwicklung grob in folgende Arbeitsschritte gegliedert:

- Menüsystem
- Pulserzeugung
- Mikrofontakterzeugung
- Decimation (Unterabtastung)
- Bandpaßfilterung
- Berechnung der Mikrofon- und Punktwolkenkoordinaten
- Laufzeitenberechnung
- Interferenzberechnung pro Punkt (Kamera-Mode) bzw.
- Interferenzberechnung pro Achse (x und y) im HID-Mouse Mode.

Da nicht damit zu rechnen war, daß die STM32F4-Timer nur 1 MHz erreichen würden, und innerhalb der HAL-Umgebung keine andere Chance sichtbar wurde, den schnellen Mikrofontakt auszugeben und die 8 Mikrofone schnell genug parallel einzulesen, entstanden einige Probleme.

Letztlich gelang es, den Mikrofontakt über eine sog. Data Watchpoint and Trace Unit (DWT) – die in den Manuals kaum beschrieben wird - in einer Art Polling-Mode zu erzeugen. Der Mikrofontakt wird mit der DWT-Unit als Viertel-Mikrosekundentakt erzeugt. (Die DWT-Unit dient normalerweise der Leistungsmessung des Prozessors).

Damit aber wurden Timing Probleme mit der USB-HID-Mouse Schnittstelle befürchtet. Deshalb wurde die Mausschnittstelle auf einem separaten Prozessor ATmega32U4 (Arduino proMicro) verlagert, der über eine UART (115200 Baud, 8n1n) jeweils 3 Byte an Daten erhält:

1. X-Verschiebung (int8)
2. Y-Verschiebung (int8) und
3. Befehlsbyte (Befehle ,r', ,l', ,d' und ,/')

Die Verschiebungen wurden bewußt nur int8 (+/- 127) gewählt. Im Befehlsbyte wird derzeit nur Linksklick übertragen. Rechtsklick und Doppelklick werden zwar vom USB-Adapter richtig weitergegeben, sind aber noch keiner Geste zugeordnet. Die Befehle sind als ASCII-Zeichen codiert:

- ,r': rechte Maustaste
- ,l': linke Maustaste
- ,d': Doppelklick
- ,/': nichts, kein Klick

Die Mikrofonbyte werden zunächst mit 2,048 MHz eingelesen. Danach werden sie auf die Abtastrate von 64 kHz gewandelt (Dezimierung um 32). Ein spezielles 22-Tap Bandpaß-Filter läßt anschließend nur eine Frequenz von  $64/3 = 21,333$  kHz durch.



```
COM3 - PuTTY
-----
Menue:
h help
a send microphone bytes as ASCII
b send samples after decimation as ASCII
c send samples after highpass as ASCII
e show mic-coordinates [mm]
i send interference values of all points
k space-set number
m USB-HID-mouse-control mode
p periodic sampling
r reset parameters to default values
s 3d-space coordinates [mm]
u calculate single delaymasks (correct with w or r)
v show version, values and parameters
w calculate double delaymasks (complete)
x send microphone bytes hexadecimal
y send samples as int32 hex

Toggle values:
d additional debug-outputs: n
m hid-mouse output: n
p periodical sampling mode: n
t mic-testdata synthesis: n
```

Abb. 18: Die Menüstruktur wurde einfach gehalten, da sie nur für Wartung oder Geräteeinstellung gebraucht wird.

Im Mode ,m' spielt das Gerät USB- Maus. In den Moden ,s' oder ,p' werden 3d-Bilder berechnet. Der periodische Mode ist zur Personenzählung oder ähnlichen Applikationen gedacht, während der space-Mode ,s' eher dem Debugging und der Verifikation der Bilder dient.

Mit der Befehlsgruppe a, b und c können die Zeitfunktionen in unterschiedlichen Bearbeitungsstufen überprüft werden. Mit den Befehlen x und y dito, jedoch in hexadezimaler Darstellung zur schnelleren Übertragung in Verbindung mit der Windows-Oberfläche.

Mit der Funktion ,e' werden die Mikrophon-Koordinaten angezeigt:

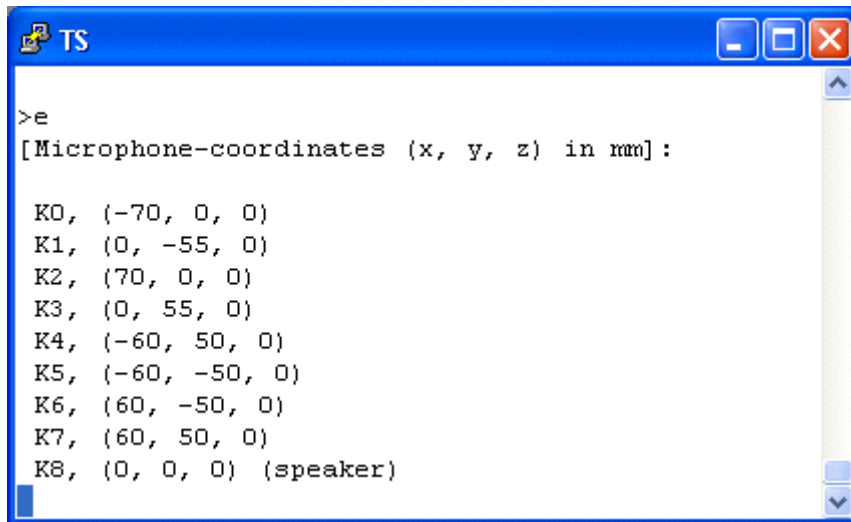


Abb. 19: Anzeige der Mikrophonkoordinaten mit ,e'

Mit der Funktion ,s' kann man sich die berechneten Koordinaten der Punkte ansehen:

```

>s
[p(i, j, k,) = (x, y, z) in mm]
(0, 0, 0), (-100, -45, 110)
(0, 0, 1), (-100, -45, 130)
(0, 1, 0), (-100, -40, 110)
(0, 1, 1), (-100, -40, 130)
(0, 2, 0), (-100, -35, 110)
(0, 2, 1), (-100, -35, 130)
(0, 3, 0), (-100, -30, 110)
(0, 3, 1), (-100, -30, 130)
(0, 4, 0), (-100, -25, 110)
(0, 4, 1), (-100, -25, 130)
.....

(40, 15, 0), (100, 30, 110)
(40, 15, 1), (100, 30, 130)
(40, 16, 0), (100, 35, 110)
(40, 16, 1), (100, 35, 130)
(40, 17, 0), (100, 40, 110)
(40, 17, 1), (100, 40, 130)
(40, 18, 0), (100, 45, 110)
(40, 18, 1), (100, 45, 130)
[p(i, j, k,) = (x, y, z) in mm]
[number of pixels: 1558]

```

Zur Überprüfung der berechneten Masken wurden die Ausgaben ,u' und ,w' implementiert. Mit ,u' werden die 8+1 Verzögerungen jedes Punktes neu berechnet und angezeigt:

```

>u
[x, y, z, k, sample_units]
0, 0, 0, 0, 23
0, 0, 0, 1, 28
0, 0, 0, 2, 39
0, 0, 0, 3, 33
0, 0, 0, 4, 28
0, 0, 0, 5, 22
0, 0, 0, 6, 36
0, 0, 0, 7, 40
0, 0, 0, 8, 29
0, 0, 1, 0, 26
0, 0, 1, 1, 30
.....
40, 18, 0, 7, 22
40, 18, 0, 8, 29
40, 18, 1, 0, 41
40, 18, 1, 1, 36
40, 18, 1, 2, 26
40, 18, 1, 3, 30
40, 18, 1, 4, 38
40, 18, 1, 5, 42
40, 18, 1, 6, 31
40, 18, 1, 7, 25
40, 18, 1, 8, 32
[x, y, z, k, sample_units]
[number of delays: x41*y19*z2*k9 = 14022]

```

Mit ‚w‘ werden die acht kompletten Zweiwege- Verzögerungsmasken (Mikrofon plus Lautsprecher) des Punktes dargestellt.

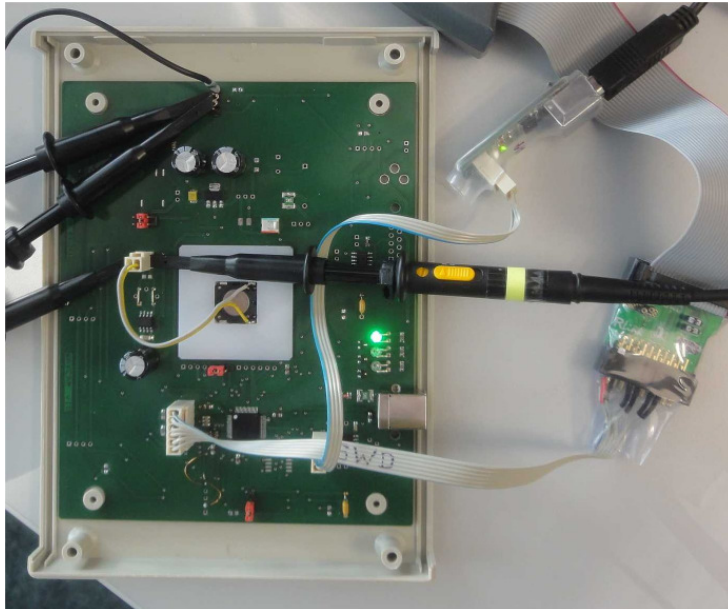
```

>w
[x, y, z, k, sample_units]
0, 0, 0, 0, 52
0, 0, 0, 1, 57
0, 0, 0, 2, 68
0, 0, 0, 3, 62
0, 0, 0, 4, 57
0, 0, 0, 5, 51
0, 0, 0, 6, 65
0, 0, 0, 7, 69
0, 0, 1, 0, 58
0, 0, 1, 1, 62
.....
40, 18, 0, 5, 69
40, 18, 0, 6, 57
40, 18, 0, 7, 51
40, 18, 1, 0, 73
40, 18, 1, 1, 68
40, 18, 1, 2, 58
40, 18, 1, 3, 62
40, 18, 1, 4, 70
40, 18, 1, 5, 74
40, 18, 1, 6, 63
40, 18, 1, 7, 57
40, 18, 1, 8, 32
[x, y, z, k, sample_units]
[number of delays: x41*y19*z2*k8 = 12464]

```

## ***Erprobungs- und Testphase***

Zur Inbetriebnahme leistete das im Projekt angeschaffte Picoscop hervorragende Dienste. Die Oszillogramme im Bericht wurden damit erstellt. Programmierung und Debugging des Prozessors erfolgten über die SWD-Schnittstelle.



*Abb. 20: Inbetriebnahme der Software. Zu erkennen sind Probes von Picoscop und Logikanalysator (Saleae), der Raisonance Rlink- Adapter für SWD und ein USB-VCP Adapter.*

Vorrangig für Zwecke der Buskommunikation, aber auch für die Ersteinrichtung des Prozessors sollten von ST angebotene Programmbibliotheken genutzt werden. ST bietet die sog. HAL-Library (Hardware-Abstraction-Layer) und eine zugehörige Software „Cube“ an. Damit begannen verschiedene Probleme. Cube erzeugt dutzende spezifische Systemfiles, deren Zusammenspiel dem Außenstehenden weitgehend unklar bleibt.

Die Ersteinrichtung der C-printf() Funktion mit der Cube-HAL ergab, daß über UART gesendet, aber nicht empfangen werden konnte. Erst nach Wochen konnte der Fehler gefunden werden. Das UART-RX Pin war durch Fehlbedienung von Cube als Input deklariert worden. Dies verhinderte die Funktion als UART-Rx.

Mit den Timern 3 und 4 des Prozessors wurde versucht, den Takt für die Mikrofone herzustellen. Dazu wurde ein Grundtakt von 4 bis 7 MHz benötigt. Intensive Arbeit ergab schließlich ein endgültiges Meßergebnis. Bis alle HAL- Unterprogramme abgearbeitet sind, vergehen rund 670 ns. Die Timer arbeiten damit maximal mit rund einem Megahertz. In Anbetracht der hohen Taktfrequenz von 168 MHz enttäuscht dies. Hunderte Internet-Forenseiten wurden recherchiert, um eine Lösung zu finden. Unter Einbeziehung der „Data Watchpoint and Trace Unit (DWT)“ [11] gelang sie. Als Alternative hätte sich ein PWM- Mode eines Timers angeboten. Leider bevorzugen PWM- Ausgänge spezifische Pins, die bereits anders vergeben waren. Die DWT-Lösung erfüllt allerdings nicht alle Wünsche, da Taktzyklen durch Rechenoperationen verloren gehen. Der Mikrofontakt variiert damit je nach Programmteil um bis zu 10%.

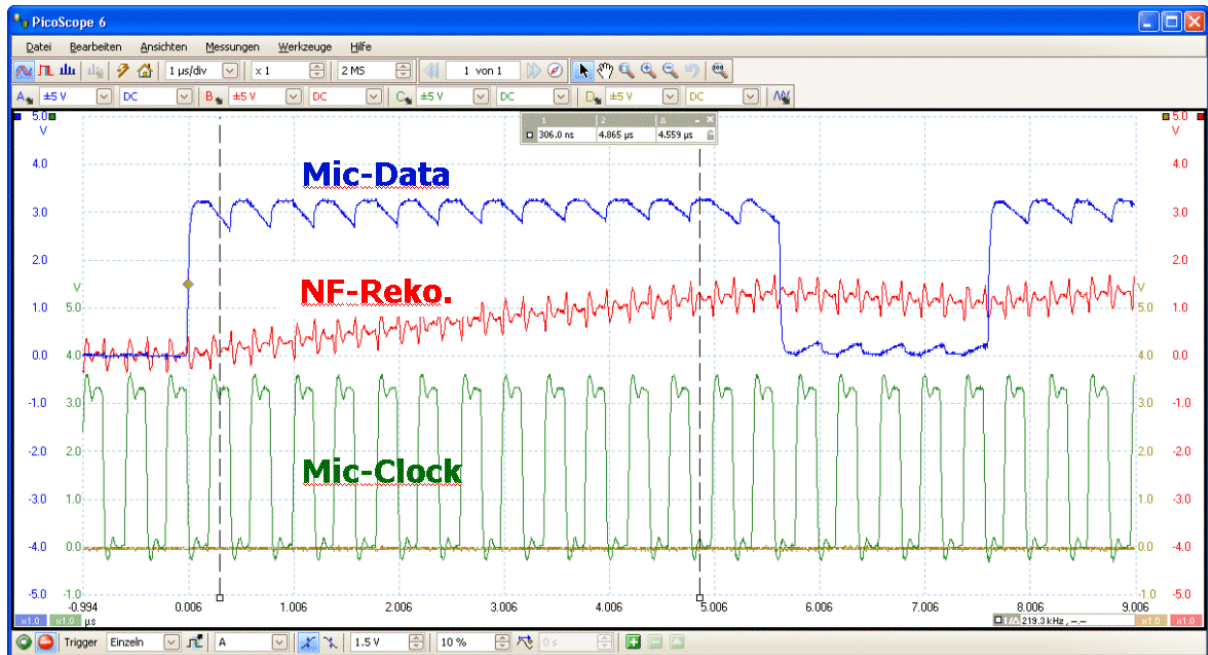


Abb. 21: PDM-Datenstrom eines Mikrofons. Zur Rekonstruktion der Niederfrequenz (NF) wurde hier ein RC-Glied (10kOhm/3,3nF) benutzt.

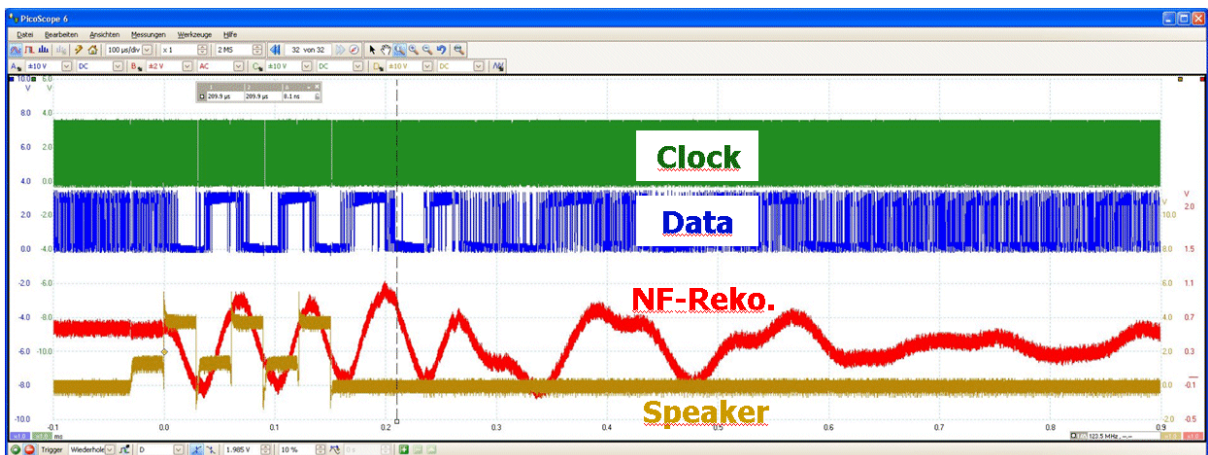


Abb. 22: Maximale Aussteuerung eines Mikrophons auf 21,3 kHz. Man erkennt an weiß bleibenden Stellen im Digitalsignal (DATA), daß der maximale Signalanstieg erreicht ist. Rot ist das rekonstruierte NF-Signal dargestellt, gelb das Signal des Lautsprechers.

Intensive Arbeit galt der Rekonstruktion der digitalen Mikrofondaten. Verwendete, digitale MEMS-Mikrophone vom Typ ST MP34DT01TR [13] liefern einen pulsdichtemodulierten, 1-Bit Datenstrom mit rund 2 MHz. Diese Daten werden zunächst um 32 auf eine Samplerate  $f_s$  von 64 kHz dezimiert. Danach schließt sich ein Bandpaß mit der Arbeitsfrequenz von 21,3 kHz an ( $f_s/3$ ).

Im Internet, wie in der Literatur sind zum Thema der Filtertheorie unzählbar viele Quellen zu finden. Leider aber zeigen alle verwendeten Mikrofone ein aussteuerungsabhängiges Offset, dessen Beseitigung ernsthafte Probleme bereitet. Zur Gestaltung von Dezimierungsfilttern, noch zur speziellen Offset-Beseitigung war keine Web- Hilfe zu erhalten.

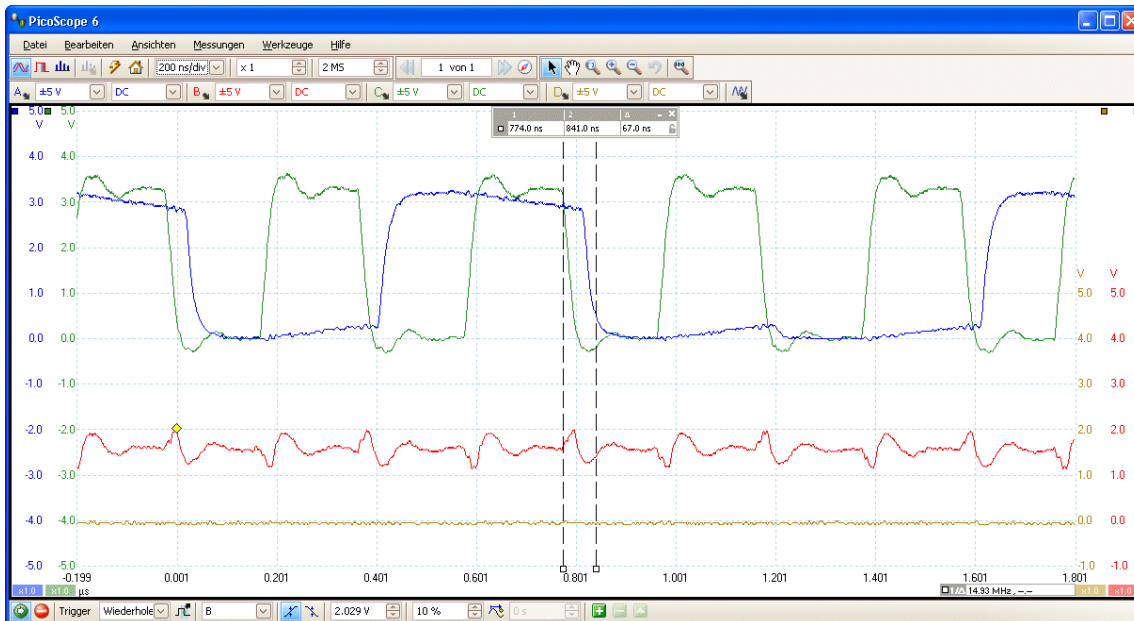


Abb. 23: Nachlaufende Data-Flanke des Mikrophons (gestrichelt). CLK: grün, DATA: blau, NF: rot.

Bei der Analyse wurde festgestellt, daß das Mikrophon mit Verzögerung auf einen HL-Flankenwechsel des Takts reagiert, siehe Abb.23. Spezifiziert sind 18 ns, gemessen werden 67 ns. Da nicht klar ist, wann der Prozessor mit einer Quantisierung von 21 MHz übernimmt, wurde die Erkennung auf den H-Zustand des Takts verlegt. Alle 8 Mikrophone sind mit LR = low beschaltet.

Dann zeigt der Betrieb der Mikrophone bei 23 kHz ein extrem starkes Rauschen. Ausflüge in die Systemtheorie liefern bislang Erklärungsansätze, aber noch keine befriedigenden Lösungen. Trotz intensiver Bemühungen gelang es nicht, dem Gesamtsystem aus Lautsprecher und Mikrophon eine Reichweite von mehr als 10 cm abzutrotzen. Danach verschwindet das Lautsprechersignal auf drei versuchten Übertragungsfrequenzen 16kHz, 21,3 kHz und 32kHz so stark im Rauschen, daß keine Reste nachweisbar sind.

Da bis zum Schluß Hoffnung bestand, einen Fehler in den Algorithmen zu finden, konnten keine alternativen Anstrengungen entwickelt werden. Final wurde klar, daß die Verwendung digitaler Mikrophone im Ultraschallbereich durchaus sehr problematisch ist.

Das Problem mangelnder Empfindlichkeit konnte bis zum Projektende nicht behoben werden. Folgende Ursachen sind denkbar:

- 1) Die Mikrophone wurden zu heiß gelötet (nicht verifizierbar, Bestückung erfolgte außer Haus)
- 2) Die Leiterplatten mit Mikrofonen wurden nach dem Löten gewaschen (für Mikrophone unzulässig)
- 3) Der Frequenzgang ist anders, als in Application Notes beschrieben
- 4) Das Rauschen der Mikrofone steigt frequenzproportional
- 5) Eine Decimation von 32 ist im Ansatz zu gering.

Um die Lautstärke des abgestrahlten Signals zu erhöhen, wurden verschiedene Varianten erprobt:

- Dynamischer Lautsprecher mit Betriebsspannungen von 5V und 15 Volt
- Piezolausprecher mit Pulstransformator

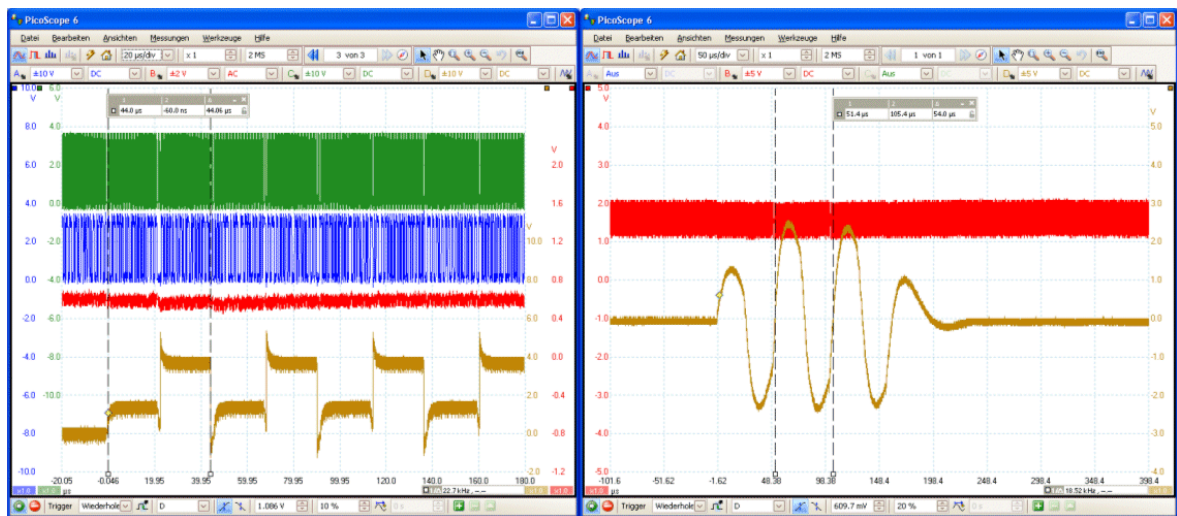


Abb. 24: Links: Zeitfunktion am dynamischen Lautsprecher (gelb). Rechts: Zeitfunktion (gelb) am Piezolausprecher mit Pulstransformator (beides mit VDD = 5V).

Es zeigt sich, daß hier verwendbare, dynamische Lautsprecher eine zu kleine Induktivität besitzen, um in die Nähe von Resonanz zu kommen, Abb.24 links. Die Sättigung wird zu schnell erreicht, danach begrenzt der ohmsche Widerstand der Spule den Strom. Entsprechend fällt eine relativ hohe Spannung über den Transistoren des Treiber- IC ab. Im Bild fallen oben wie unten etwa 1,3 Volt über den Transistoren ab. Es fließt Kurzschluß-Strom, die Betriebsspannung gerät in Gefahr, abzufallen.

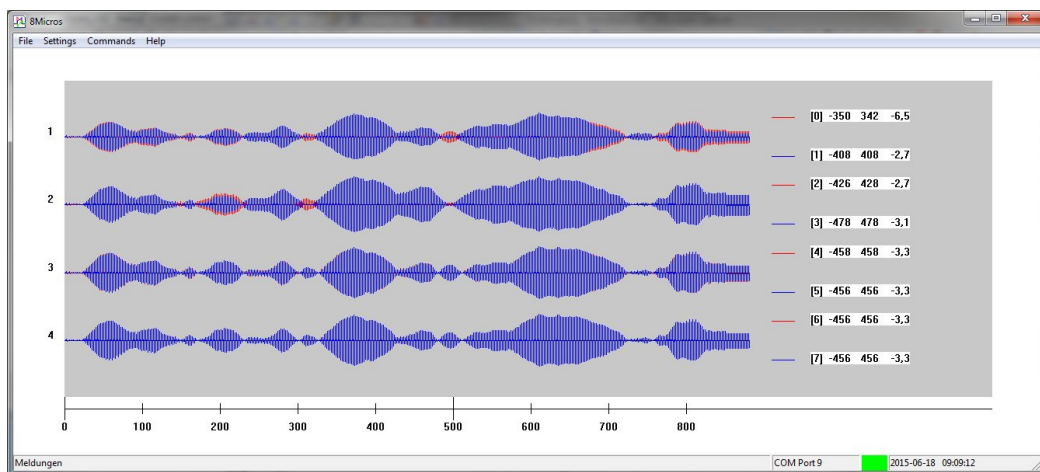


Abb. 25: Visualisierung der Mikrophondaten unter Windows7. Dargestellt sind vier mal zwei Kanäle (je in rot und blau).

Im Gegensatz dazu zeigt die Variante mit Pulstransformator und Piezolausprecher (rechts) Resonanzverhalten. Hier ist es möglich, mit wesentlich geringeren Strömen sogar noch eine höhere Wirkung zu erzielen.

Um entwickelte Programme testen und erproben zu können, wurde ein Programm entwickelt, mit dem Mikrofondaten und Bilder unter Windows geprüft werden können.

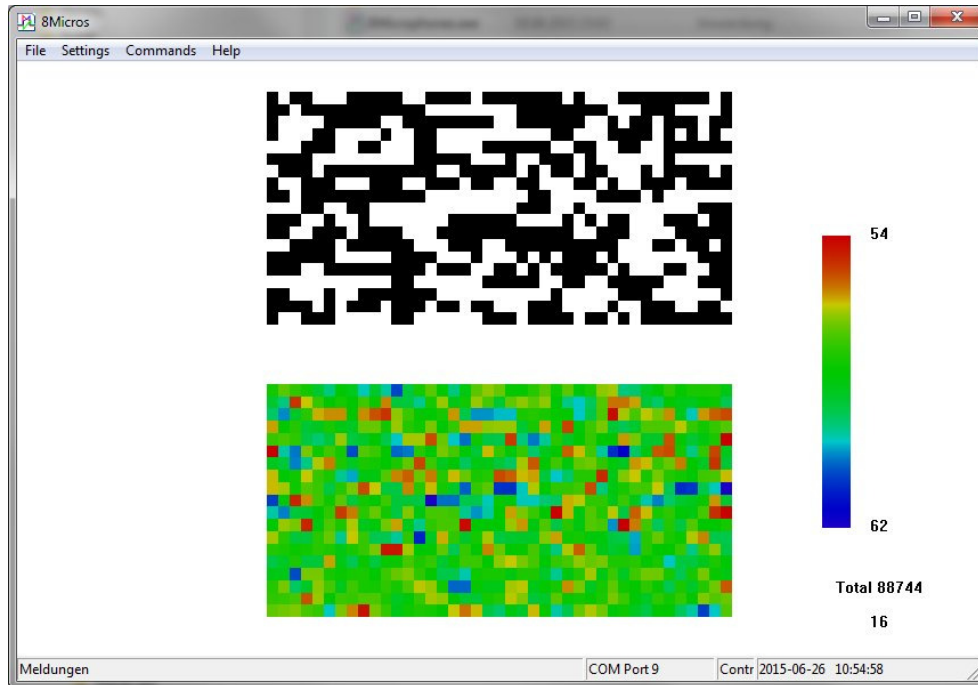


Abb. 26: Visualisierung eines Ergebnisbildes ohne Echo.

Abb. 26 zeigt die Darstellung eines Ergebnisbildes. Unten ist das Bild der z-Ebene maximaler Immission dargestellt, während oben eine Ebenenzuordnung der Immissionen pro Pixel dargestellt ist. Dabei bedeutet weiß die z-Ebene 0 und schwarz die z-Ebene 1. Die höhere Immission wird angezeigt.

Die Farbskala rechts paßt sich automatisch dem Maximum und dem Minimum des unteren Bildes an. Unter der Farbskala ist die im Bild steckende „Energie“ als Summe aller (immer positiven) Interferenzwerte zu erkennen. Darunter ist die Anzahl gemachter Aufnahmen zu erkennen, hier ‚16‘. Das Programm dient ausschließlich der Einrichtung und Fehlererkennung.

Das Bild gestattet eine Abschätzung der Rauschkennwerte der Mikrofone, wie auch eine Abschätzung der Immissionen.

### **3 Bewertung der erzielten Ergebnisse in Gegenüberstellung mit den Zielsetzungen des Antrages, Bezugnahme auf die Notwendigkeit und Angemessenheit der geleisteten Arbeit, Bezugnahme auf die wichtigsten Positionen des zahlenmäßigen Nachweises**

Mit der Förderung dieses Projekts durch BMWi gelang es, Erkenntnisse auf neuartigen Wissensgebieten zu gewinnen. Einerseits konnte die Möglichkeit der ultra-kompakten Realisierbarkeit von Ultraschallkameras und touchless Displays erstmals untersucht werden, andererseits war eine Entschlüsselung der Daten von pulsweitenmodulierten, digitalen MEMS- Mikrofonen durch Software Neuland. Nicht zuletzt entstand aus dem Wunsch heraus, eine einfachste und perfekte Dauermodulation für Ortungen mit Wellen zu finden, ein Zugang zur Signalcodierung mit Rauschfunktionen, deren Kartierungseigenschaften dem des Einzelimpulses entsprechen sowie eine zugehörige Bauanleitung für Schlüssel (Zeitfunktion) und Schloß (FIR-Filter).

#### ***AP1: Entwicklung Simulationssystem***

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Entwicklung eines Simulators (Scilab)
- Erstellung eines Funktionsgenerators
- Optimierung des Netzwerkmodells
- Optimierung der Zeitfunktionen und Parameter

Ausgangspunkt für das Arbeitspaket war die Fragestellung, inwiefern „unkomplizierte“ Zeitfunktionen (z.B. mit Werten -1, 0, 1) benutzt werden können, die gegenüber einem Dirac- Puls mehr Signalenergie mitführen (continuous wave, CW-Verfahren), und die zugleich mit geringstem Soft- und Hardwareaufwand dekodierbar (faltbar) sind. Auch soll deren Faltung mit sich selbst einen scharfen Spike ohne Artefakte liefern. Es entstanden Scilab- Simulationsprogramme zur Klärung der Eigenschaften rauschartiger, bandbegrenzter Zeitfunktionen und deren Faltungen. Zeitfunktionen und deren sphärische Realisierungen im Netzwerkmodell (Interferenznetz) wurden untersucht. Zeitfunktionen und Parameter wurden diskutiert.

Zur Realisierung geräuschfreier Sendesignale wurden insbesondere bandbegrenzte Rauschsignale untersucht, bei denen die Überföhrungsfunktion  $H$  bandbegrenzt und arm an Anteilen unterhalb von 20 kHz ist. Neben der Untersuchung von Rauschsignalen wurden Spektrum und Faltung von Barker-Codes untersucht, siehe auch [2..4].

Es entstand ein Simulationssystem mit Zeitfunktionsgenerator zur Berechnung des Interferenzsystems. Netzwerkmodell, Zeitfunktionen und Parameter der Wellenkamera wurden berechnet und optimiert.

#### ***AP2: Simulation des Interferenzsystems***

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Netzwerk-Simulation des Ultraschallsystems
- Optimierung der akustischen Parameter
- Minimierung der Rekonstruktionsschrittzahl
- Variation von Frequenz- und Zeitparametern

Faltungsoperatoren und deren Netzwerkrealisierung wurden untersucht. Dabei ging es um den Nachweis von Eigenschaften von Faltungsintegralen (Faltungssummen). Es konnten Relationen

zwischen Ortssummen, Zeitsummen und gemischten Summen (Faltungssummen) erkannt werden. Dabei wurde eine Javascript-Homepage weiterentwickelt [1], die gewonnene Erkenntnisse festhält.

Zur Ersparnis von Rechenzeit wurden „einfache“ Zeitfunktionen gesucht. Bei der Untersuchung bipolarer und unipolarer Zeitfunktionen und deren Faltungen wurde eher zufällig entdeckt, daß die Faltung einer unipolaren Welle mit einem bipolaren Gewichtsmuster ohne DC-Artefakte möglich ist. Die Bedeutung dieses überraschenden Ergebnisses ist nicht nur für das Projekt, sondern auch für die Neuroforschung interessant, erlaubt es doch zukünftig vielleicht, komplexe Muster auch im Nervensystem zu verstehen.

Da ein ursprünglich im Projekt vorgesehenes analoges Ultraschall-Mikrofon (Knowles SPM0404UD5) seit Projektbeginn nicht mehr lieferbar ist und keine brauchbaren Alternativtypen existieren, mußte das Systemkonzept verändert werden.

Die Ablösung durch Piezo- Transducer begrenzt den möglichen Spielraum für Sende-Zeitfunktionen. Verschiedene Variationen der Interferenzrekonstruktionen mit unterschiedlichen Sensor- und Aktoreigenschaften wurden durchgespielt. Eine Vielzahl von Datenblättern verfügbarer IC sowie verfügbarer Sensoren und Aktoren wurden geprüft.

Akustische Parameter für Piezotransducer und Mikrofone wurden im Wellenlängenbereich um 40 kHz (8,5mm) untersucht. Dabei zeigt sich, daß die starke Richtcharakteristik und große Resonanzgüte Probleme in den Bereich der Algorithmik induzieren. Die Kapseln müssen stark bedämpft werden, um pseudostochastische Zeitfunktionen (rauschartige) ohne Amplitudenbeeinflussung übertragen zu können. Die hohe Güte von Piezo- Transducern fordert Spikes oder periodische Signale – beide sind für den zu erfüllenden Zweck nicht optimal geeignet. Eine Interferenzrekonstruktion mit dem (Heinzschen) Maskenalgorithmus leidet in der Simulation unter starken Amplitudenschwankungen und Nebengeräuschen.

Zur Optimierung der akustischen Parameter wurden Varianten entwickelt. Für Piezo- Transducer und analoge Mikrophone wurde ein Modell mit folgenden akustischen Eigenschaften (0 dB = 20 µPa) simulativ und in Teilen experimentell verifiziert:

- Reflektierte Weglänge 4 m (+/-2 m) ~ 12 ms
- Mehrfach-Pulse mit 40 kHz
- abgestrahlter Sendepiegel SPL > 90 dB auf 30cm bei 10V RMS
- Schalldämpfung bei 40 kHz um -5 dB/m
- Objekt- Reflexion -10 ... -40dB
- Abtastrate 80...96 kHz
- Empfangssignalstärke variiert dann von 30 dB bis 80 dB (632 µPa bis 0,2 Pa)
- Bei Mikrofonempfindlichkeit von 1 mV/Pa (worst case) folgt die benötigte Eingangsempfindlichkeit des Vorverstärkers zu 0,632 µV ... 0,2mV
- Dies ist zugleich der Regelbereich eines analogen Vorverstärkers
- Ein Sender, vier bis acht Empfänger
- Signalfilterung digital, Detektion der Einhüllenden
- Rekonstruktion durch Effektivwert der Delay-Maske über der Einhüllenden
- Zwischenspeicherung im RAM
- Bei 96 kS/s für 12 ms: 1200 Samples mal Kanalzahl mal ADC-Bits (8/16: 150 kB)
- Zyklisches Senden einer Pulsgruppe alle 20 ms

### **Simulation des Interferenzraumes**

Bei der Netzwerk- Simulation des Ultraschallsystems konnten als Zielparameter fixiert werden: Samplerate 64 kHz, Sendefrequenz 25 kHz, Objektabstand 0,5 ... 3,5 Meter, Speichertiefe 882 Samples ~ 28244 Miples (Mikrofonbyte), Zeitauflösung (Sampletakt) 15,6µs, Ortsauflösung 5,3mm. Das Optimum der Winkelauflösung gilt stets nur für eine Entfernung. Tabelle 1 zeigt erreichbare Winkelaufösungen  $\alpha$  in Abhängigkeit von Objektentfernung  $r$ , auf den Sampleabstand normierter Objektentfernung  $n$  und Mikrofonabstand  $a$ . Aus der multikriteriellen Wahl wurde eine optimale Winkelauflösung für 2 Meter Objektentfernung gewählt, dabei stehen die Mikrofone etwa 140 mm auseinander.

**Tabelle 1: Winkelauflösungen des Arrays**

r [mm]	n [ ]	$\alpha$ [°]	a [mm]
200	38	13	46
500	94	8,2	29,7
1000	188	5,9	102,6
2000	376	4,17	145,8
3000	565	3,4	178,2

Die entsprechenden Koordinaten des Mikrofonarrays sind dann:

**Tabelle 2: Abgeleitete Mikrofonkoordinaten**

Kanal	x	y	z	
0	-70	0	0	mic
1	0	-55	0	mic
2	70	0	0	mic
3	0	55	0	mic
4	-60	50	0	mic
5	-60	-50	0	mic
6	60	-50	0	mic
7	60	50	0	mic
8	0	0	0	speaker

Zur Klärung des Zusammenhangs zwischen systemtheoretischer Transferfunktion (H) und Netzwerktopologie als FIR-Filter (bestehend aus Delays T und Weights W) des akustischen Interferenznetzes wurden einfache Algorithmen entwickelt. Sie wurden in neuronaler Ausprägung im Konferenzbeitrag [2] veröffentlicht.

### **AP3: Entwurf der $\mu$ C-Schaltungstechnik**

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Auswahl eines geeigneten DSP/Mikrocontrollers ( $\mu$ C)
- Schaltungsentwicklung, digital
- Schaltungsentwicklung, analog
- Verifikation eines Ultraschall-Zyklus

Die einschränkende Signalformung von Piezo- Transducern führte zur Frage nach alternativen Lösungsvarianten. Bei Verwendung von NF-Mikrofonen mit nachzuschaltendem Bandfilter entstehen Lösungsmöglichkeiten mit je einem Ultraschall-Lautsprecher und 4 bis 8 MEMS- Mikrofonen, sowohl in analoger, als auch digitaler Form.

Zunächst war hier die Frage der Störunterdrückung im NF-Bereich (100 Hz bis 20 kHz) zu klären. Zufällig wurde bei ersten Experimenten mit der Kapsel MP45DT02 (im Evakit „Discovery“ des zu verwendenden Prozessors STM32F4) bemerkt, daß dieses MEMS- Mikrofon eine im Datenblatt nicht vermerkte Resonanz im Bereich von 20 bis 40 kHz aufweist. Mit Überhöhungen von bis zu 14 dB ist zu rechnen. Da es Hinweise darauf gibt, daß auch andere MEMS- Kapseln dieses in den Datenblättern nicht vermerkte Verhalten zeigen, sind wahrscheinlich verschiedene (analoge und

digitale) MEMS- Mikrophone als Ultraschallsensoren geeignet. Im Gegensatz zu Transducern können bei Verwendung von MEMS rauschartige Zeitfunktionen sowie Barker – Codes verwendet werden, die Signalenergie läßt sich steigern, ohne die Lautstärke zu erhöhen.

Analoge Lösungsvarianten, bestehen aus analogem MEMS-Mikrophon, Signalbegrenzer, Bandfilter, Vorverstärker und Analog-Digitalwandler (ADC) wurden untersucht. Für höchste Flexibilität sorgt ein Ansatz mit analogen MEMS. Dabei sind zusätzliche, einstellbare Vorverstärker erforderlich. Akustische Eckkennwerte sind vergleichbar zum Transducermodell. Dieser Variante wurde viel Aufmerksamkeit gewidmet, letztlich aber entsteht mit ca. 25 Bauelementen pro Mikrofonkanal ein recht hoher Hardwareaufwand. Unerwartete Probleme bereitet auch die Wahl steuerbarer Vorverstärker (Gain Controlled Amplifier GCA), eine diskrete Realisierung dieser aus digitalem Potentiometer und zwei Operationsverstärkern hätte den Aufwand nochmals erhöht. Verfügbare GCA weisen eine grobe Amplitudenrasterung aus, z.B. AD8231 sieben Schritte, Verdopplung der Verstärkung je Schritt. Diese kann zu algorithmischen Problemen führen, wenn innerhalb einer Maske die Verstärkung verdoppelt wird.

Eine weitere Variante sollte die Verwendbarkeit von hochauflösenden Audio- ADC (CS5368) klären. Wird die Auflösung des ADC von 12 Bit auf 24 Bit erhöht, so steigt die Amplitudenauflösung um den Faktor  $10 \text{ Bit} = 1024$ . Aus der Stufung  $3,3\text{V} / 12 \text{ Bit} = 0,8 \text{ mV}$  werden so  $0,8 \mu\text{V}$  am Eingang. Entsprechend entfallen steuerbare Vorverstärker. Leider aber zeigt sich, daß auch diese Variante im Detail ebenfalls nur mit hohem Hardwareaufwand zu realisieren ist, der einer optimalen Vermarktung im Wege stünde.

Wird ein schneller Mikrocontroller vorausgesetzt, so kommt eine digitale Lösungsvariante ins Blickfeld. Für geringsten Hardwareaufwand sorgt ein Systemkonzept mit einem Ultraschall- Aktor sowie acht digitalen MEMS- Mikrofonen, die direkt an einen Port des Mikrocontrollers angeschlossen sind. Der im digitalen MEMS enthaltene Sigma- Delta Wandler erzeugt einen 1-Bit PDM- Datenstrom mit 1...3,5 MHz. Wird ein 8 Bit breiter Port benutzt, sind daran z.B. 16 Mikrophone direkt anschaltbar. Die Taktversorgung der Mikrophone kommt vom Mikrocontroller. Die Maximalamplitude von MEMS liegt oberhalb von 120 dB(A), ist also relativ unkritisch. Zur Eigenrauschgrenze der Mikrophone liegen allerdings keine nachvollziehbaren Daten vor. Verfügbare Mikrophone unterscheiden sich im Verhältnis des Anstiegs der Rauschleistung zum Amplitudengang. Wird ein Rechenweg analog zum Rechengang der Piezo- Transducer gewählt, ergeben sich unkritische Verhältnisse. Probleme bereitet allerdings das Timing beim Einlesen, siehe AP3. Hierfür stehen maximal 56 Maschinentakte zur Verfügung, mit denen die Daten gelesen und in 32 Bit gestopft werden müssen.

Im Detail wurden mit dem Ziel der Minimierung von Volumen und Fertigungskosten, aber auch der Minimierung des Entwicklungsaufwandes sowie der Optimierung des Gesamtverfahrens verschiedene Schaltungslösungen für Empfangs- und Sendebaugruppen untersucht:

- Analoger Mikrophoneingang (für Elektret- und analoge MEMS- Mikrophone)
- Digitaler Mikrophoneingang für digitale MEMS (MP45DT02, MP34DT01)
- Hochauflösender, achtkanaliger 24-Bit ADC (CS5368 mit 8-chl I<sup>2</sup>S) für analoge Mikrophone
- Lautsprecher-Verstärker und Ansteuerung in Vollbrücke für Mikrophon- Varianten
- Transformatorisches Transducer- Design mit kombiniertem Aktor/Sensor (AD8231)
- Trafofreies Transducer- Design mit kombiniertem Aktor/Sensor (AD8231)
- Klassisches Beamforming- Design mit Chipsatz (MAX8082, -81, -79; LM965xx etc.)
- Diskrete Realisierung eines geregelten Vorverstärkers mit AGC
- Verwendung von Spezial-OPV mit AGC (SSM2167 etc.)

Eine Bewertung der verschiedenen Kombinationen ergab folgendes Bild:

- Teuerste und aufwendigste Variante stellen klassische Beamforming- Chipsätze aus der medizinischen Ultraschalltechnik dar, 7).
- Zur Verfügung stehende IC stellen immer einen Kompromiß dar. Diskrete Schaltungstechnik 8) aber wird bereits bei vier Kanälen unakzeptabel voluminös.
- Als universellste Variante zeigen sich 3) in Verbindung mit 4). Dennoch ist der Hardware-Aufwand beachtlich. Problematisch sind Variable Gain Controlled Amplifier, einsetzbare IC sind zu grobgeteilt (z.B. AD8231: 7-stufig), hier ist mit algorithmischen Problemen bei der

Verstärkungsumschaltung zu rechnen. Andererseits sind Lösungen mit konventionellem OPV und Potentiometer zu aufwendig.

- Vandalismussicher sind die Transducer- Varianten 5) und 6), allerdings mit dem Nachteil eingeschränkt verwendbarer Zeitfunktionen.
- Als preiswerteste und kleinste Variante stehen 2) und 4) zur Verfügung. Neben dem Mikrocontroller ist nur ein Taktpuffer und ein Lautsprecherverstärker erforderlich, insgesamt also nur 3 IC.

Teilschaltungen wurden entwickelt, berechnet und simuliert. In Wechselwirkung mit der algorithmischen Weiterentwicklung fiel schließlich die Entscheidung, die Variante mit digitalen Mikrofonen zu favorisieren. Diese ist zwar mit dem höchsten Risiko behaftet, da noch keine Erfahrungen mit diesem neuen Mikrofontyp bestehen, dennoch vermittelt die Handy-Branche das Bild, daß diese Mikrophone funktionieren können.

Um hinreichend lange Datensätze zwischenspeichern zu können, wurde ein MRAM Typ MR25H40 integriert. Damit stehen 4 Mbit = 0.5 Mbyte mehr Speicher zur Verfügung, der mit 40 MHz bewegbar ist, siehe auch AP5.

### ***AP4: Leiterplattenentwicklung***

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Fixierung der Systemparameter
- Festlegung der Bauelementespezifikationen
- Festlegung der Schaltungstechnik
- Leiterplattenentwurf
- Versuchsaufbau

Im Arbeitspaket wurde eine erste, digitale Version entwickelt, die mit digitalen, als auch mit Zusatzadapter mit analogen Eingängen bestückbar ist. Eingänge können mit digitalen MEMS-Mikrofonen bestückt werden. Damit ist die Baugröße des Prototyps später minimierbar. Der Hardwareaufwand ist minimal. Im Falle von Software-Problemen in der Erprobungsphase können dennoch Anpassungen erfolgen. Die Ausgabe des Moduls erfolgt mittels UART, RS485 und USB. Damit sind die zwei möglichen Zielvarianten (Personenzähler im Bus, bzw. Touchless Display am PC) mit unterschiedlichen Adaptern am anderen Ende bedienbar.

### ***AP5: Programmentwicklung $\mu$ C***

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Installation/Inbetriebnahme Compiler  $\mu$ C
- Modularisierung und Spezifikation
- Variation der Prozessorspezifik
- Programmentwicklung Sender
- Programmentwicklung Empfänger

Systemkonzept, sowie Konfiguration der Module wurden in Interaktion mit der Hardwareentwicklung begonnen. Die Einarbeitung in den neuen Prozessor stellte die größte Hürde dar. Mit dem Wachstum des Integrationsgrades werden die Prozessoren immer komplexer und Beschreibungen immer unvollständiger.

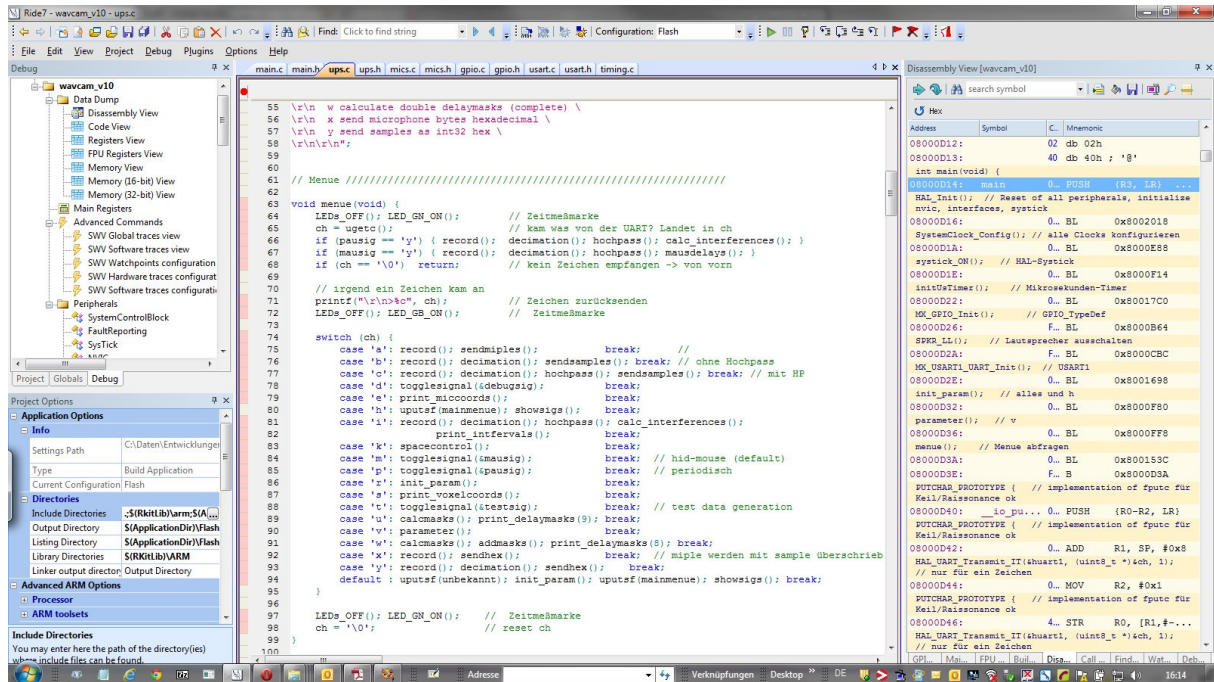


Abb. 27: Raisonance Ride7 Oberfläche mit GNU-C Compiler im Debug-Mode, rechts am gelben Codefenster zu erkennen

Bei der algorithmischen Umsetzung waren Besonderheiten der gewählten Variante mit digitalen Mikrofonen zu lösen:

- 1) Im Zeitraum der Datenaufnahme stehen je  $168\text{MHz} / 3\text{MHz} = 56$  Prozessortakte zur Verfügung, um die Daten aller Mikrophone zu lesen und in 32-Bit-Pakete zu sortieren. Während dieser Arbeit sind Interrupts für etwa 12 Millisekunden zu sperren. Damit ist eine Ausführung komplexer Busprotokolle nicht unkompliziert. Es mußte ein Mikrocontroller gefunden werden, der die unterstützten Schnittstellen in Hardware abbildet, soll nicht ein separater Kommunikationsprozessor entwickelt werden. Die Wahl fiel auf den STM32F405 mit USB und CAN in Hardware.
- 2) Nach Aufnahme der Daten kann die Verarbeitung der Daten vergleichbar gemacht werden. Da der Prozessor nur 192 kB (CCM + SRAM1 + SRAM2) besitzt, wurde der Schaltungsentwurf ergänzt mit einem 512 kByte MRAM, siehe AP3. Fallen pro Mikrophon und Meter Schallweg rund 8,8 kBit  $\sim 1,1\text{kByte}$  an, können mit 512 kByte rund 500 Mikrophonmeter zwischengespeichert werden, bei vier Mikrofonen ist die Reichweite auf  $512 / 4 / 2 = 64$  Meter begrenzt, bei 8 Mikrofonen sind 32 Meter Schallweg aufzeichnenbar.

Aus 1) und 2) folgend wird eine Zwischenspeicherung im RAM mit je 8 Bit vorgenommen. Das RAM wird bei 8 Mikrofonen mit 2 MB/s beschrieben.

Im Ergebnis entstand ein für den Prozessor STM32F405 optimiertes C-Programm, welches

- den Prozessor initialisiert (diverse Programme),
- Variablen und Konstanten anlegt,
- das Menüsystem erzeugt menu(),
- die Koordinaten der zu berechnenden Punkte des Interferenzraumes bestimmt voxelcoords(),
- die Abstände der Punkte zu den Mikrofonen und zum Lautsprecher bestimmt calc\_masks(),
- die Masken auf Samples umrechnet und Lautsprecherentfernung addiert add\_masks(),
- mit der Maske den Interferenzwert jedes Punkts bestimmt cal\_interferences() oder
- die Mausspur verfolgt mauscontrol()

## AP6 Programoptimierung

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Auslagerung kritischer Codeteile in Assembler
- Funktionsoptimierung
- Geschwindigkeitsmaximierung
- Einbeziehung applikativer Aspekte

Die gewählte HAL- Entwicklungsumgebung des Cortex-M4 Prozessors schließt den Anwender weitgehend davon aus, Modifikationen an Bibliotheken vorzunehmen. Einerseits ist sie sehr komplex, andererseits ist die Machart teilweise unverständlich. Deren Anlage gleicht der eines monströsen Betriebssystems. Überall dort, wo es auf Geschwindigkeit ankommt, versagt die Verarbeitung. So sind die UARTs mit nicht mehr als 115200 Baud betreibbar, die Ausgabe des Mikrofon-Takts gelingt in verschiedenen Varianten nur mit 1 MHz (4...7 MHz sind erforderlich). Pro Ausgabe eines Wertes wurden bis zu 670 ns gemessen. Im AP waren viele, ungeahnte Probleme zu lösen, deren Ursachen in der Komplexität des Prozessors, wie in der Komplexität der HAL Bibliotheken lag.

Eine Auslagerung kritischer Codeteile in Assembler war nicht nötig. Die Funktion des Gesamtsystems wurde hergestellt. Insbesondere die Geschwindigkeit der Mikrofon-Portzugriffe konnte optimiert werden. Der Speicherplatz von Mikrofon-Bytes (Miples) und Samples wird jetzt doppelt genutzt.

## AP7 Entwicklung einer Beispielapplikation

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Problemanalyse
- Umsetzung auf Wellenkamera
- Programmierung
- Codeoptimierung
- Programmierung des Human Interface

Beide im Projektantrag benannten Beispielapplikationen wurden entwickelt. Eine USB-HID Ausgabe ermöglicht die Übernahme der Maussteuerung eines PC. Damit ist es über den ab Windows7 angebotenen „PC-Eingabebereich“ möglich, den PC nur mittels WAVCAM zu steuern.

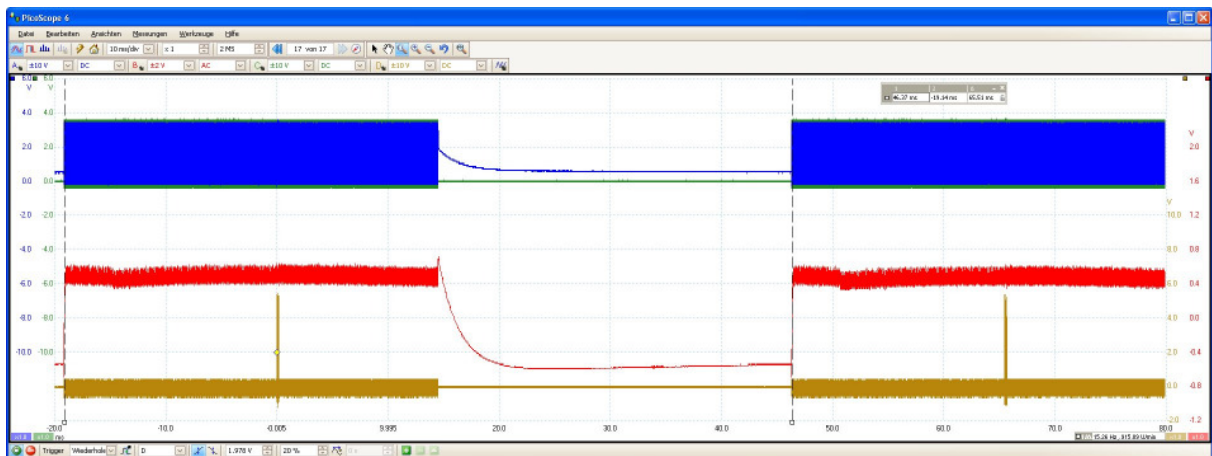


Abb. 28: *Periodischer Mouse-Mode ,m'. Gelb: Signal des Lautsprechers als Spike, Rot: Analogsignal; blau und grün: Daten und Takt der Mikrofone. Das Intervall mißt genau 65,5 ms ~ 15,3 Hz. Der Takt wird etwa 15 Millisekunden vor dem Klick begonnen (nach Spec. der Mikrofone).*

Im Mausbetrieb wird eine Abarbeitungsfrequenz von 15 Hz ~ 66 ms erreicht.

Die Voraussetzungen für einen Einsatz zur Personenzählung konnten geschaffen werden. Im bildgebenden Mode ‚p‘ entwickelt WAVCAM innerhalb von etwa 53 Millisekunden 1558 Bildpunkte (41\*19\*2). Die erreichte Abarbeitungsfrequenz von 19 Hz genügt den Anforderungen.

## AP8 Systemerprobung

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Erprobung in Anwendungsumgebung
- Untersuchung und Festlegung von Schnittstellen
- Optimierung des Gesamtsystems
- Systemüberarbeitung, Iteration

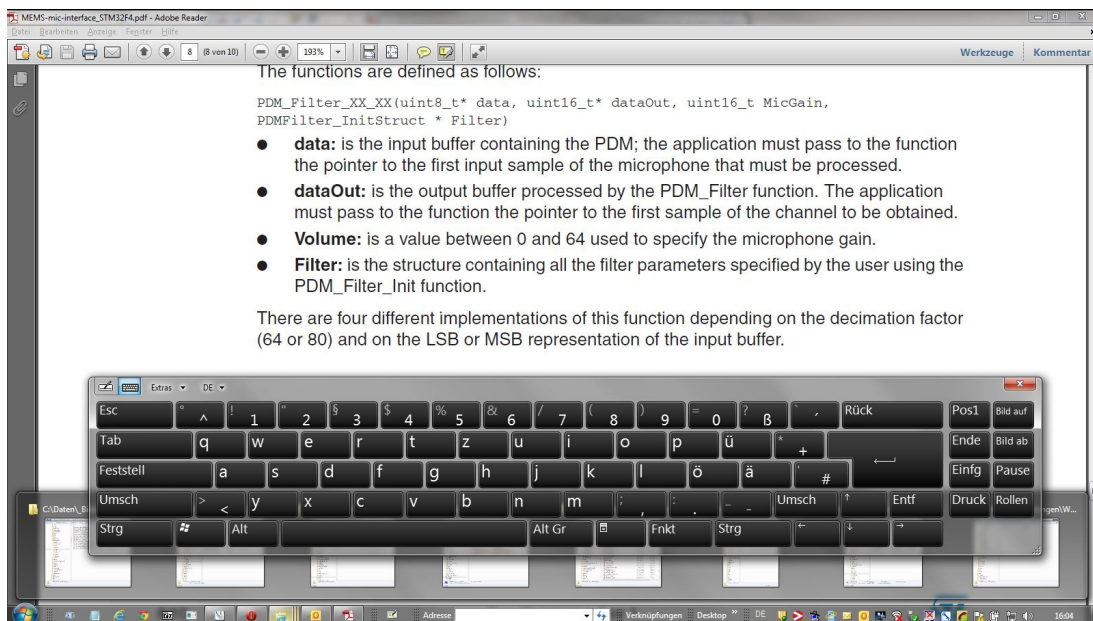


Abb. 29: Bedienungversuch des „Tablet-PC-Keyboards“ von Microsoft. Ein Bedienungsproblem zeigt sich, sobald in der Taskleiste unten ein Symbol gedrückt wird (hier der Explorer). Je nach Einstellungen ist dann entweder die Tastatur oder der Explorer „oben“.

Leider konnten die Fragen zur unzureichenden Mikrofonempfindlichkeit der verwendeten digitalen MEMS-Mikrophone [13] bis zum Projektende nicht erschöpfend beantwortet werden. Im Abschnitt 5 wird deutlich, daß offenbar auch große Firmen derzeit auf ein empfindliches Ultraschall- Mikrofon warten.

Im AP wurden deshalb verschiedene Tests erdacht, um das System dennoch optimal testen und soweit wie möglich erproben zu können. So wurde statt der Lautsprecher- Reflektion am Objekt ein Lautsprecher am Kabel durch das Bildfeld geführt. Die Erprobung der Mausfunktionen erfolgte zunächst über eine Terminal-Fernsteuerung, danach wurden Erprobungen mit Lautsprecher vorgenommen. Es zeigt sich, daß derzeit vom Lautsprecher abgestrahlte Geräusche (ein Test erfolgte von 21 bis 32 kHz) noch hörbar sind. Unter Umständen sollte bei einer Überarbeitung die Abstrahlungsfrequenz auf 40 kHz erhöht werden.

Das System wurde in einen Zustand versetzt, daß die Arbeit bei Erscheinen geeigneter, digitaler Mikrofone sofort abgeschlossen werden kann.

Das größte Entwicklungsrisiko des Projekts bestand darin, mit im Prozessor verfügbaren Rechenleistungen und RAM-Größen nicht auszukommen. Mit der Systemerprobung konnte gezeigt werden, daß in beiden Arbeitsmoden (HID-Mouse und 3d-Scan) Abtastfrequenzen größer 15 Hertz erreichbar sind. Einsatzmöglichkeiten sind damit gesichert. Dieser Nachweis gelang mit dem Projekt. Mit Arbeitsfrequenzen von 15 bzw. 19 Hertz sind beide angestrebten Applikationen schnell genug.

## **AP9 Dokumentation**

Im Arbeitspaket sollten folgende Aufgaben bearbeitet werden:

- Patentrecherche, evtl. Patentierung
- Veröffentlichungsarbeit
- Abschlußbericht
- Resultat-Homepage

Im Rahmen einer online Patentrecherche wurden keine das Verfahren belastenden Patente gefunden, wohl aber Veröffentlichungen, die einer Patentierung im Wege stehen (Samsung, Nokia), siehe Abschnitt 5. Eine Patentierung wurde auch deshalb nicht forciert, weil Erprobungsergebnisse zu schwach ausfielen. Der Abschlußbericht liegt hiermit vor. Homepages für wesentliche Projektergebnisse wurden gestaltet, und ins Internet gestellt [1...4]. Eine Homepage für das Projekt soll erst hochgeladen werden, wenn das im Moment noch nicht lösbare Problem mangelnder Mikrofonempfindlichkeit beseitigt ist.

## **4 Darstellung der Innovationspotentiale und Applikationsmöglichkeiten**

- Wissenschaftliche und wirtschaftliche Bedeutung, Anwendungspotential
- Anwendungsbereiche in der mittelständischen Wirtschaft
- Darlegung der Ergebnisverwertung (eigene Nutzung, Technologietransfer, Know-how-Verkäufe u.a.)
- Darlegung der Applikationsmöglichkeiten für die mittelständische Industrie
- Perspektive und Chancen für sich anschließende Entwicklungsarbeiten

Neben der Phasen-, Code- und Frequenzmodulation wird die im ersten AP entwickelte Rauschmodulation als spezifische Form einer Codemodulation für Kartierungsfragen zukünftig große Bedeutung erlangen. Sie setzt die Fragen, die Barker bis zu Code Nr.13 beantwortete, ins Unendliche fort. Allein mit diesem Teilergebnis werden schon bald dutzende neuer Ortungsverfahren für Radar, Akustik oder Laser entstehen. Da sehr große FPGA existieren, ist es ein leichtes, faltende FIR-Filter nahezu beliebiger Größe zu entwickeln. Diese können einerseits den Schlüssel als trinäre Zeitfunktion ausgeben, andererseits wirken sie selbst als Schloß. Für Applikationen mit WLAN, GPS, Radar oder Sonar entstehen interessante, neue Möglichkeiten. Rauschartige Zeitfunktionen zeichnen sich durch schlechteste Erkennbarkeit für denjenigen aus, der ihren Code nicht kennt. Andererseits maximieren sie die Energie für Applikationen, bei denen die Spitzenleistung begrenzt ist (praktisch alle). Zukünftig werden Radargeräte nur kontinuierlich Rauschen. Wird der Rauschzyklus so lang gewählt, wie die Laufzeit für maximale Entfernung, entstehen keine Trugbilder. Damit sind Radarsender auch kaum noch ortbar. Überhorizontadar wird mit wenigen Kilowatt statt mit 10 Megawatt möglich.

Auch kann man sich mit dem Ergebnis des AP1 vorstellen, eine breitbandige Lichtwelle im Photo holografisch einzufrieren. Wird diese nun invertiert, entsteht ein rauschender, ultrakurzer pseudo- Puls für optische Meßzwecke. Dieser braucht keinerlei erregenden Peak, braucht nur winzige Dauerleistung und kommt ganz ohne Wärmeentwicklung daher. Hier gelang eine Entdeckung am Rande, die großes Zukunftspotential in verschiedenen Industriebereichen besitzt.

### **3d- touchless- Displays**

Mit dem Projekt konnte gezeigt werden, daß ultraschallbasierte touchless Displays in einer extrem platzsparenden Weise machbar sind.

Neben dem Team arbeiten auch Samsung, Nokia, Apple und Microsoft an 3d-touchless Displays, wobei einerseits Ultraschall-Lösungen (Samsung, Nokia) andererseits Kamera-basierte Lösungen in Entwicklung sind (Microsoft, Apple), siehe Abb.30 und Abb.31. Auch sind touchless Displays im öffentlichen Bereich interessant zum Beispiel bei der Bahn als Fahrplanauskunft, an städtischen Info-Points, als Bank-Terminals, oder als Eingabegerät für Behinderte.

### **3d-Scanner**

Applikationen sind im Bereich Personenzählung, intelligente, bewegungserkennende Warnsysteme und medialer Eingabegeräte (Alternativen zu Kinect) zu erkennen. Verschiedene Firmen, wie [5] zeigen Interesse.

In Anbetracht der Negativ-Schlagzeilen, die Videokameras derzeit in Abhöraffaires machen, bietet sich Ultraschall überall dort an, wo es darauf ankommt, kamerafreie, abhörsichere Bedienung durchzusetzen.

Die Wavcam selbst braucht noch eine gewisse Reife. Zunächst sind empfindlichere Ultraschallmikrofone zu besorgen (jetzt, zum Projektabschluß wurden vier neue Typen auf den Markt gebracht – leider zu spät). Wie im Antrag geplant, ist die Echo- Reichweite auf drei Meter zu erhöhen. Die Arbeitsfrequenz ist auf 40 Kilohertz zu steigern, da der Mensch auch die 32 kHz noch wahrnimmt. Hier wird ein komplettes Redesign von Mechanik, Hardware und Software nötig. Dafür werden große Anstrengungen zu unternehmen sein. Entsprechend ist die Abtastrate auf 100 kHz zu erhöhen. Der Mikrofontakt betrüge bei digitalen MEMS bei einer Decimation von 32 genau 3,2 MHz. Es erscheint aber derzeit nicht sicher, ob digitale Mikrofone die geforderte Leistungsfähigkeit bei einer Decimation von 32 überhaupt erreichen können. Unter Umständen ist es klüger, einen Schritt rückwärts zu gehen, und eine Wellenkamera mit analogen US-Mikrofonen und Vorverstärkern zu entwickeln.

Die mit dem Projekt erreichte Ausgangslage ist hervorragend, da alle nötigen Meß- und Experimentierwerkzeuge sowie ein Grundwissen geschaffen werden konnten.

Einsatzzwecke der Wellenkamera werden ganz sicher bei touchless Displays und bei 3d-Meß- und Detektionssystemen aller Art liegen. Sofern noch ausstehende Probleme mit der zu geringen Empfindlichkeit/Reichweite des Gesamtsystems überwunden sind, stehen interessante Applikationen offen.

## **5 Angaben zu erworbenen bzw. angemeldeten Schutzrechten für Vorhabensergebnisse**

Eine abschließende Patentrecherche erbrachte Neuigkeiten. Die größte davon ist, daß Samsung bereits für das Galaxy S4 ein vierkanaliges 3d-Ultraschall-Display vorsah, a). Aus welchen Gründen es noch nicht zur Verfügung steht, ist unbekannt, siehe Abbildung. Obwohl das Bild im Internet zu finden ist, ist eine zugehörige Patentnummer nicht auffindbar.

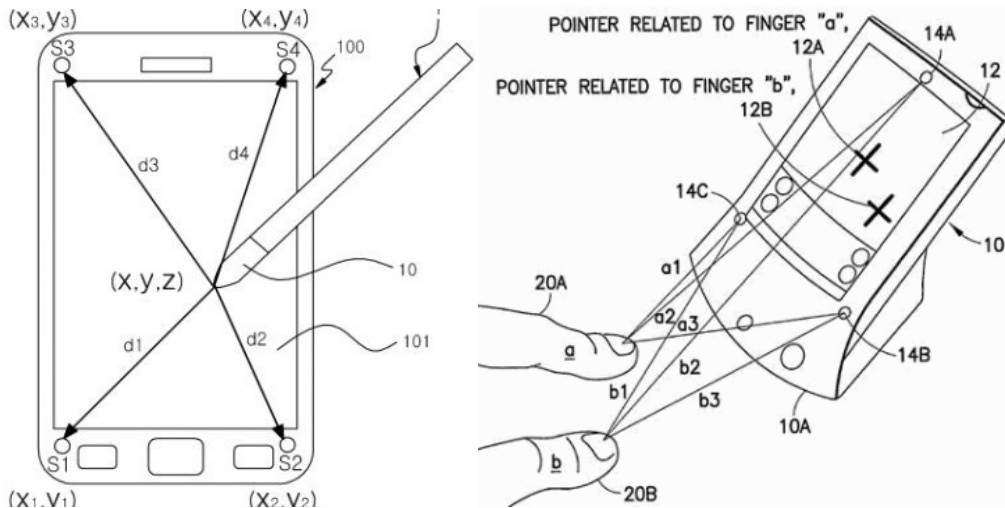


Abb. 30: Links: Vierkanaliger Entwurf für das Smartphone Samsung Galaxy S4 (3-dimensional). Rechts: Nokia-Entwurf (3-kanalig)

- a. Samsung patent could mean the death of the Galaxy Note digitizer. <http://tabtec.com/android/new-samsung-patent-could-mean-the-death-of-the-galaxy-note-digitizer/>

Nokia arbeitet offenbar seit Jahren an einem 3-Kanal System, siehe Bild. Im Bild ist ein veraltetes Handy zu erkennen. Das kann zweierlei heißen: Entweder, man entwickelt seit geraumer Zeit an dieser Technik oder die Infos werden gegeben, um falsche Fährten zu legen. So ist bekannt, daß alle Smartphone- Hersteller an Gestenerkennung arbeiten. Diese lassen sich derzeit besser mit VIS-Kameras erkennen, dort ist der Entwicklungsstand weiter.

- b. Nokia S60 Phones Could Feature 3D Touchless Control. <http://news.softpedia.com/news/Nokia-S60-Phones-Could-Feature-3D-Touchless-Control-76072.shtml>

Microsoft und Apple dagegen scheinen Lösungen der Bildverarbeitung zu favorisieren. Nach dem Erfolg mit Microsofts Kinect scheint man diesen Weg weiterzugehen.

- c. Microsoft Patent Reveals a Highly Advanced Touchless Input System for Future Mobile Devices & Beyond. <http://www.patentlymobile.com/2014/04/microsoft-patent-reveals-a-highly-advanced-touchless-input-system-for-future-mobile-devices-beyond.html>

Ein sehr einfaches dreikanal 3d-Ultraschallsystem für Hobbyisten kommt von der Cornell-University. Hier werden absolute Laufzeiten gemessen, dafür genügen drei Kanäle.

- d. Karl Gluck, David DeTomaso: UltraMouse 3D - spatial human interface that interact in three dimensions. Cornell University. [https://courses.cit.cornell.edu/ee476/FinalProjects/s2009/kwg8\\_dmd54/kwg8\\_dmd54/index.html](https://courses.cit.cornell.edu/ee476/FinalProjects/s2009/kwg8_dmd54/kwg8_dmd54/index.html)

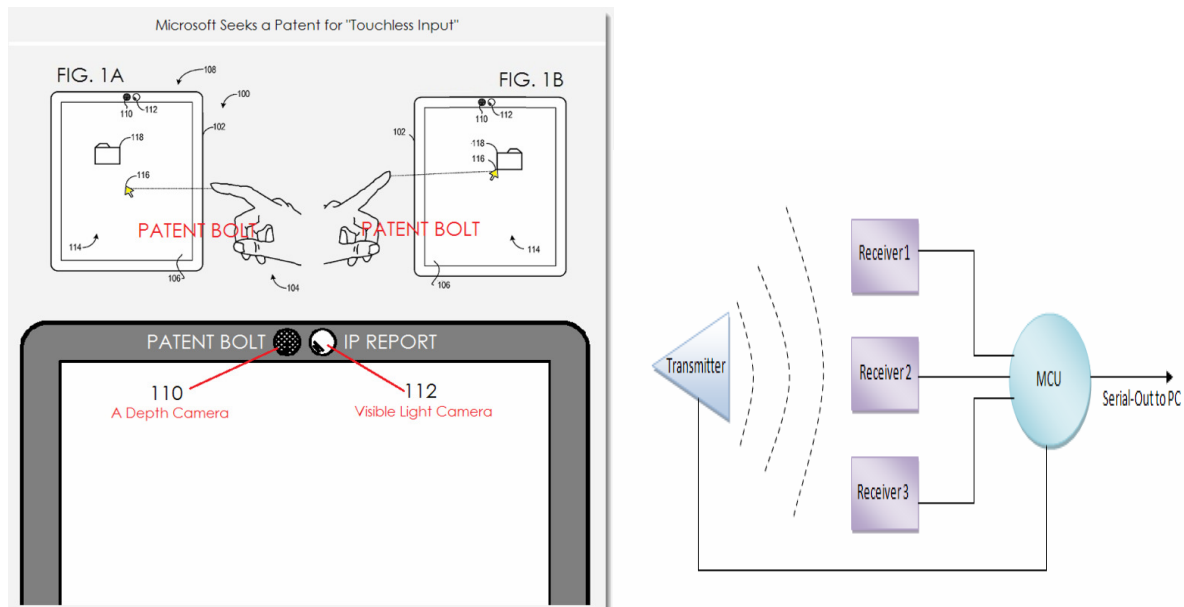


Abb. 31: Links: Microsofts Intention für Smartphones orientiert sich am Kinect. Die Überschrift des Bildes ist interessant. Man nutzt eine Tiefenkamera und eine VIS-Kamera. Rechts: Das Hobbysystem ‚Ultramouse-3d‘ der Cornell University nutzt einen in der Hand zu haltenden Sender.

Insbesondere das Interesse von Samsung und von Microsoft verrät, daß die Wavcam- Arbeiten recht weit vorn liegen. Wenn man an Smartphones denkt, wird bewußt, daß außer den Mikrofonen nicht der geringste Platz zur Verfügung steht.

Der mit Wavcam gewählte Entwurf unter Verwendung digitaler MEMS- Mikrophone wäre auch im Smartphone- Segment unumgänglich. Geht man davon aus, daß die Entwicklungsarbeiten bei Samsung und Nokia etwa zeitgleich oder früher als bei uns anfangen, könnte das darauf hindeuten, daß man auch hier noch Probleme mit der mangelhaften Empfindlichkeit der Mikrophone im Ultraschallbereich hat. Dies erklärt eventuell auch, warum zum Projektende vier neue Ultraschallkapseln aufgetaucht sind.

Würde eine erfolgsorientierte Anschlußfinanzierung zum Projekt vorhanden sein, könnten wir dank fertiger Algorithmen und Technik sofort am Ball sein, sobald ein geeignetes US-Mikrofon auftaucht.

Eine kontinuierliche Patentrecherche brachte keine weiteren, die Entwicklung beeinflussenden Fakten hervor. Neben einer unüberschaubaren Vielzahl von Patenten im Medizinbereich existieren Ansatzpunkte in folgenden Offenlegungsschriften:

- e. David Prater, Stephen Watkins, David Frank Adams: Simultaneous ultrasonic viewing of 3d volume from multiple directions. US 20140358004 A1. 13. Febr. 2012, <http://www.google.com/patents/US20140358004>
- f. Mckee D Poland, Martha G. Wilson: Portable 3D ultrasound system, US 7141020 B2. 20. Febr. 2002, <http://www.google.com/patents/US7141020>
- g. T.J.Nohara, P.Weber, R. Bernardi: 3d ultrasonic imaging aparatus. US 2003163046 A1. Jun28, 2003, <http://www.google.tl/patents/US20030163046>

Die Ausführungen von Samsung und Nokia besitzen hohes Potential.

Die im Projekt angestrebten Reichweiten wurden noch nicht erreicht. Mit den Ergebnissen wurden deshalb noch keine Schutzrechte angemeldet. Das Projekt war komplizierter als vermutet. Die zur Verfügung stehende Zeit reichte nicht aus.

## 6 Zusammenstellung aller erfolgten bzw. geplanten Veröffentlichungen (Artikel in Zeitschriften, Seminare, Schulungen, Vorträge, Messen, Ausstellungen, Präsentationen) und Literaturstellen

- [1] Heinz, G.: Beziehungen zwischen Faltungs- und Interferenzintegral als Verbindung zwischen Systemtheorie und Wellenoptik. <http://www.gfai.de/~heinz/publications/animations/convolution.htm>
- [2] Heinz, G.: How Network Topology Defines its Behavior - Serial Code Detection with Spiking Networks. Workshop „Autonomous Systems“, Cala Millor, Mallorca, 13.-17. Oct.2013 publ. in Fortschritt-Berichte VDI, Reihe 10, Nr.827, Unger/Halang „Autonomous Systems 2013“ [www.gfai.de/~heinz/publications/papers/2013\\_autosys.pdf](http://www.gfai.de/~heinz/publications/papers/2013_autosys.pdf).
- [3] Heinz, G.: Noise Radar. How to use continuous time functions with maximum energy and flattest spectral distribution to get sharpest impulse response. Web-page with Scilab-sourcecode examples <http://www.gfai.de/~heinz/techdocs/tnoiseconv.sce>
- [4] Heinz, G.: Joy of Convolution. Scilab- sourcecode with examples see [http://www.gfai.de/~heinz/techdocs/joy\\_of\\_convolution.sce](http://www.gfai.de/~heinz/techdocs/joy_of_convolution.sce)
- [5] Iris-GmbH: Time of Fligth Camera 'Irma Matrix'. <http://www.irisgmbh.de/produkte/irma-matrix>
- [6] Datenblätter und Applikationsnoten des Prozessors STM32F405 siehe <http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577?sc=stm32f4>  
<http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577/LN1035>
- [7] Wasserdichte, wartungsfreie Ultraschallsensoren: <http://www.murata.com/en-us/products/sensor/ultrasonic/proof>
- [8] Datenblätter Ultraschall-MEMS microphone Knowles: <http://www.digikey.de/product-detail/de/SPM0404UD5/423-1086-1-ND/1587388>
- [9] Barker-Code siehe Wikipedia <https://de.wikipedia.org/wiki/Barker-Code>
- [10] Christian Wolff: Radar-Tutorial. <http://www.radartutorial.eu/>
- [11] Liebler, Klaus: Microseconds Delay for STM32 durch Nutzung der Data Watchpoint and Trace-Unit (DWT), 15.08.2014. <http://klaus4.blogspot.de/2014/09/microseconds-delay-for-stm32.html>
- [12] Datenblatt Microsoft Kinect for Windows: [https://readytogo.microsoft.com/en-us/Asset/ layouts/Download.aspx?SourceUrl=https://readytogo.microsoft.com/en-us/Asset/RTG%20Anonymous%20Asset%20Library/08%20K4W%20Kinect%20for%20Windows\\_Technical%20Datasheet.pdf](https://readytogo.microsoft.com/en-us/Asset/ layouts/Download.aspx?SourceUrl=https://readytogo.microsoft.com/en-us/Asset/RTG%20Anonymous%20Asset%20Library/08%20K4W%20Kinect%20for%20Windows_Technical%20Datasheet.pdf)
- [13] Datenblätter verwendeter Mikrophone : [http://www.st.com/web/en/catalog/sense\\_power/FM125/SC1564/PF250940](http://www.st.com/web/en/catalog/sense_power/FM125/SC1564/PF250940)
- [14] ST Microelectronics: AN3998 - PDM Audio Software Decoding on STM32 Microcontrollers. [http://www.element14.com/community/servlet/JiveServlet/downloadBody/50350-102-1-260376/STMicroelectronics.Application\\_Note.pdf](http://www.element14.com/community/servlet/JiveServlet/downloadBody/50350-102-1-260376/STMicroelectronics.Application_Note.pdf)