

3D-Messung im
elektrischen Feld per
Influenzstromzeitintegral

Inhalt

Verzeichnis aller Abbildungen.....	2
Historie dieser Doku	2
Vorwort.....	3
Verwendete Hilfsmittel	3
Grundlagen	4
Influenzstrommessung	5
Der Influenzstrom ist die 1. Ableitung der Feldstärke	6
Wirkung des Influenzstromes auf das zu messende elektrische Feld.....	7
Rotationsvoltmeter vs. Integratormessmethode	9
Fehlkonstruierte Influenzstrommessgeräte.....	9
Realisierung.....	10
Elektrodeninterface	10
Gesamtschaltung	11
Software	12
Aufbau.....	17
Messungen.....	18
Anwendungen.....	19

Verzeichnis aller Abbildungen

ABBILDUNG 1: SENSOR IM ELEKTRISCHEN GLEICHSPANNUNGSFELD.....	4
ABBILDUNG 2: BEGINNENDER INFLUENZSTROM.....	5
ABBILDUNG 3: MAXIMALER INFLUENZSTROM	5
ABBILDUNG 4: ABKLINGEN DES INFLUENZSTROMES	5
ABBILDUNG 5: INTEGRATION DES INFLUENZSTROMES	6
ABBILDUNG 6: VERSUCHSAUFBAU ZUR FELDSTROMMESSUNG.....	7
ABBILDUNG 7: FELDSTROMVERLAUF	7
ABBILDUNG 8: FELDSTROM OHNE ROTATION	8
ABBILDUNG 9: FELDSTROM MIT ROTATION.....	8
ABBILDUNG 10: FELDSTROMMESSUNG.....	8
ABBILDUNG 11: STROMGESTEUERTE STROMQUELLE	10
ABBILDUNG 12: GESAMTSCHALTBILD	11
ABBILDUNG 13: SOFTWARE-LISTING E3D.C	16
ABBILDUNG 14: STECKBRETTAUFBAU.....	17
ABBILDUNG 15: MESSDATEN.....	18

Historie dieser Doku

03.11.2019	Initialversion
bis 08.11.19	Kleine Ergänzungen, Umformulierungen, Tippfehler

Vorwort

Angeregt durch zwei Foren reifte beim Verfasser (Wolfgang aka „Rumgucker“) die Idee zur Entwicklung eines neuartigen Geräts zur Messung elektrischer Felder.

Das Messgerät soll folgende Vorgaben erfüllen:

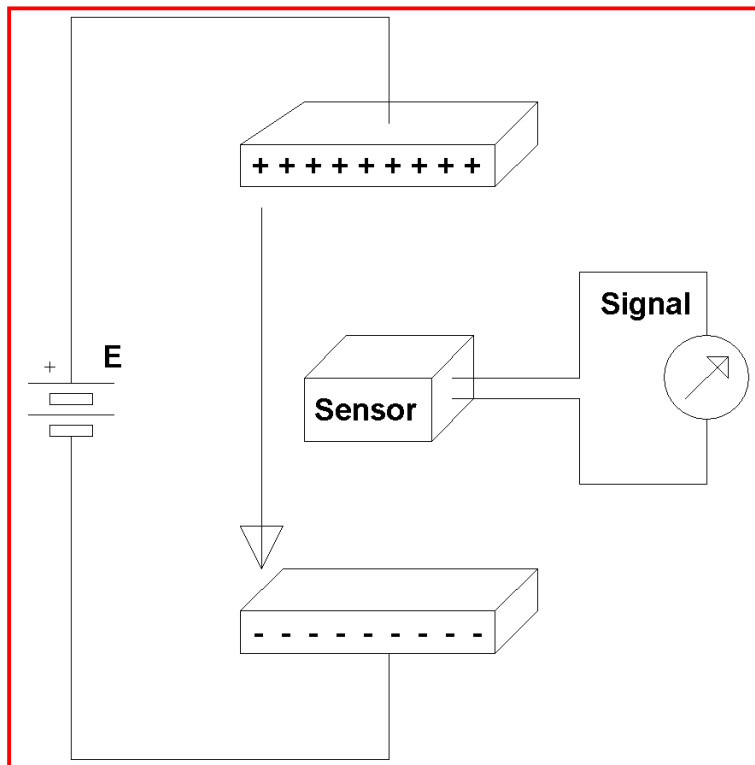
- Ungewohnter Ansatz, denn Nachkochen ist was für Langweiler
- Offenlegung des gesamten Projektes
- Praxistaugliches solid-state-Messgerät
- Einfacher Aufbau, Teilekosten unter € 10,--
- Vermessung des elektrischen Feldes in drei Dimensionen
- Datenerfassung mit Smartphone, Tablet oder PC

Diese Doku ist keine Bauanleitung. Sie soll ebenso wenig eine Einführung in die Hard- und Softwareentwicklung darstellen. Nachfolgende Seiten richten sich an erfahrene Entwickler und sollen zu ähnlichen Arbeiten und Offenlegungen anregen.

Verwendete Hilfsmittel

- ATTiny85-Entwicklungsumgebung AVR Studio 4.18
- Programmiergerät GALEP 4 für ATTiny85
- Programm LT Spice 4.23 zur Skizzen- und Schaltplanerstellung
- Sprint-Layout 6.0 zur Konstruktion von Platinen und Elektroden
- CNC-Fräsmaschine zur Fertigung von Platinen und Elektroden
- Datenblätter, Netzteile, Oszillographen, Multimeter, Steckbrett, PC, Tablet.

Grundlagen



Im Labor kann ein variables elektrisches Gleichspannungsfeld durch zwei voneinander entfernte Feldplatten erstellt werden, die mit einer Gleichspannungsquelle verbunden sind. Das Feld ist ein Vektor – hier mit einem Pfeil dargestellt. Die Feldstärke wird in $E = U / d$ [V/m] angegeben.

Das Ausgangssignal eines zu entwickelnden Sensors soll diese Feldstärke anzeigen.

Abbildung 1: Sensor im elektrischen Gleichspannungsfeld

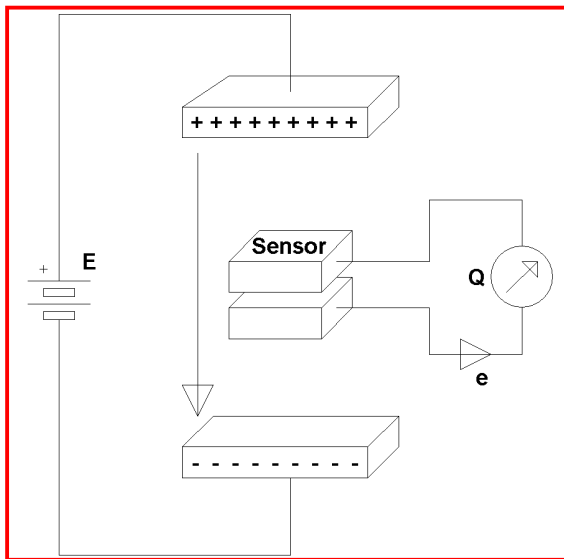
In der Natur könnte die obere Feldplatte eine geladene Gewitterwolke und die untere Feldplatte der Erdboden sein.

Folgende Vorrichtungen zur Messung elektrischer Felder sind zum Beispiel bekannt:

- Blättchenelektroskop
- Ein- und Zweifaden-Elektrometer
- Braunsches Voltmeter
- Kelvin Voltmeter
- Rotationsvoltmeter (ugs. „Feldmühle“)
- Elektrofeldmeter

Viele klassische Geräte messen die Kraft, die elektrische Ladungen ausüben. Sie zeigen also die Spannung an. Man kann über die Spannung bei bekannten Dimensionen des Messgeräts auf die gesuchte Feldstärke zurückrechnen.

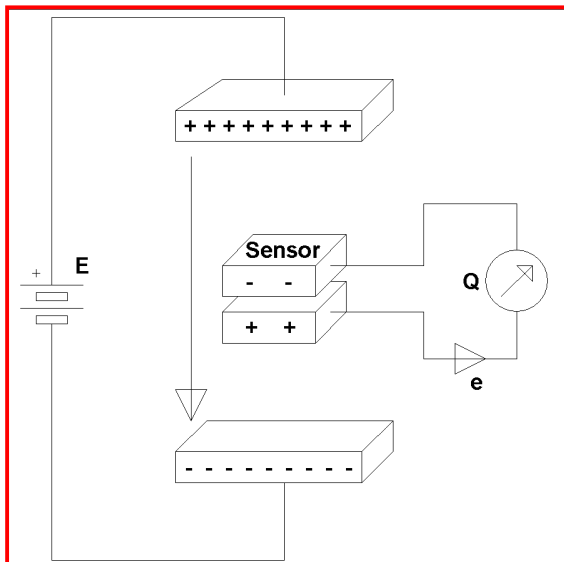
Influenzstrommessung



Anders arbeiten Geräte, die den Influenzstrom messen. Die beweglichen Ladungsträger der anfänglich ungeladenen Sensorplatten werden von dem einwirkenden elektrischen Feld verschoben.

Die Elektronen wollen die untere Sensorelektrode verlassen, da sie von der unteren Feldelektrode mit ihren negativen Ladungsträgern abgestoßen werden. Und sie werden von der oberen Feldelektrode mit dem Elektronenmangel angezogen.

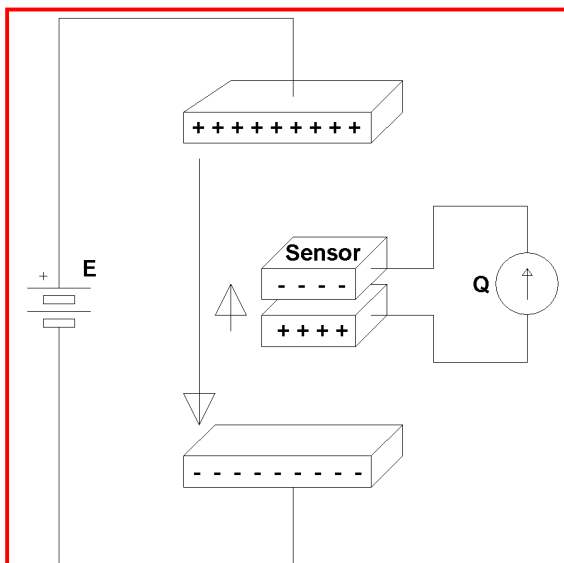
Abbildung 2: Beginnender Influenzstrom



Die obere Sensorplatte lädt sich durch die hinzukommenden Elektronen zunehmend negativ, während auf der unteren Sensorplatte Elektronenmangel herrscht.

Das anliegende elektrische Feldes führt zu einer Ladungsverschiebung, die die elektrische Feldstärke im Raum zwischen den beiden Sensorplatten auszugleichen versucht.

Abbildung 3: Maximaler Influenzstrom



Wurden genug Elektronen verschoben, so stellt sich im inneren Bereich zwischen den beiden Sensorplatten ein feldfreier Raum ein und der Influenzstrom klingt ab.

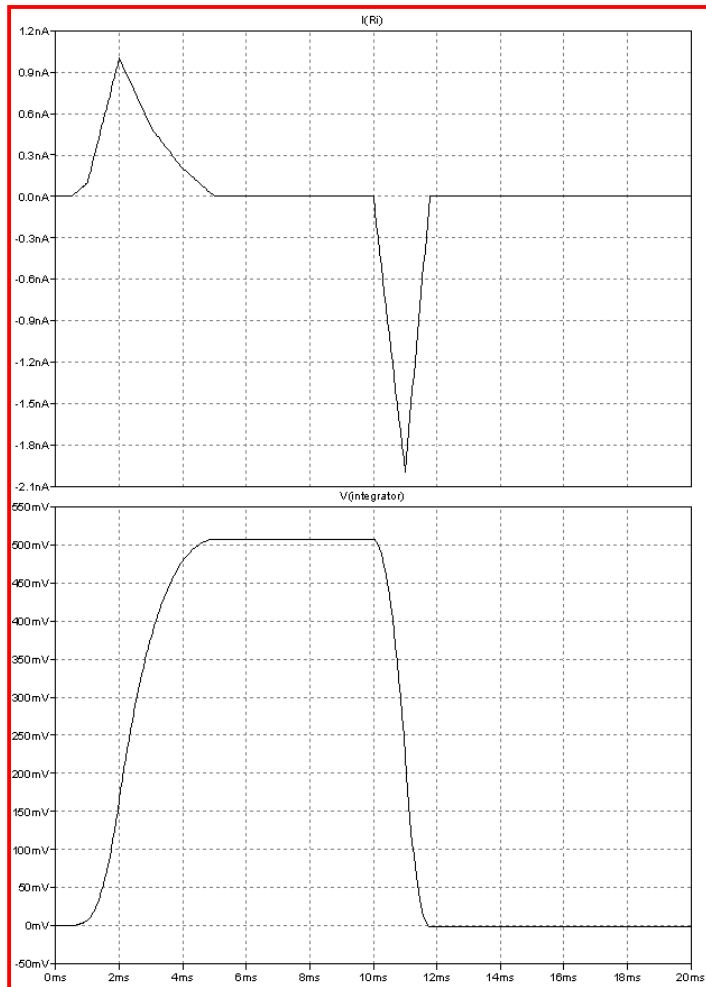
Der faradaysche Käfig arbeitet nach diesem Prinzip.

Die verschobenen Ladungsträger werden danach durch die Kraftwirkung des einwirkenden elektrischen Feldes an ihrer Position gehalten.

Abbildung 4: Abklingen des Influenzstromes

Der Influenzstrom ist die 1. Ableitung der Feldstärke

Ein Influenzstrom fließt, sobald sich das Feld, das auf die Sensorplatten einwirkt, ändert.



In der linken Messkurve bedeutet ein positiver Influenzstrom eine Zunahme des von den Sensorplatten erfassten Feldes. Und ein negativer Influenzstrom bedeutet eine Abnahme des erfassten Feldes. Kein Influenzstrom bedeutet, dass sich das erfasste Feld nicht ändert.

In der Grafik wurden die Sensorplatten langsam in das Feld eingetaucht und nach kurzer Verharrung schnell wieder aus dem Feld entfernt. Man hätte alternativ auch die Spannung des Feldes verstellen können.

Integriert man dieses Strommesssignal, so erhält man die zu jedem Zeitpunkt wirkende Feldstärke.

Abbildung 5: Integration des Influenzstromes

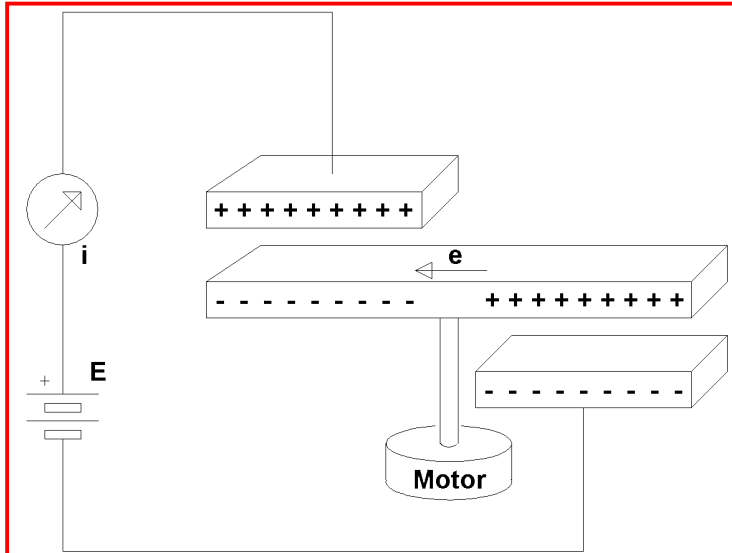
Ein Messgerät zur Ermittlung der elektrischen Feldstärke besteht aus einem Strommessgerät und einem nachgeschalteten Integrator.

Hiermit sind relative Messungen möglich. Alle Feldmessungen beziehen sich auf die Feldstärke am Ort der Inbetriebnahme des Integrators. Wenn man absolute Felder messen will, so muss man den Sensor in einem faradayschen Käfig in Betrieb nehmen und danach den Käfig entfernen. Dann zeigt das Gerät auch die Feldstärke von Gewitterwolken an.

Sehr langsame Feldänderungen oder sehr schwache Felder bewirken nur geringe Influenzströme, die ggfls. nicht mehr erfasst werden können. Auch werden schnelle Wechselfelder (z.B. durch das Wechselstromnetz) elektromagnetisch ausgemittelt und können nicht gemessen werden.

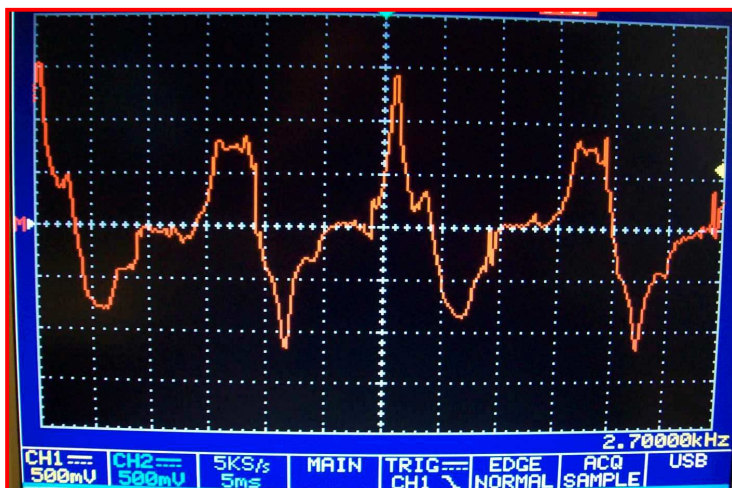
Wirkung des Influenzstromes auf das zu messende elektrische Feld

Bei jeder Änderung des von den Sensorplatten erfassten Feldes fließen Influenzströme zwischen den Sensorplatten. Es genügt dazu das Eintauchen oder Entfernen der Sensorplatten aus dem elektrischen Feld. Die Frage ist, ob dann zugleich auch im Feldkreis ähnliche Ströme nachweisbar sind.



Der Versuchsaufbau besteht aus einer rotierenden länglichen Sensorelektrode, die von dem an den Enden anliegenden Feld zweimal pro Umdrehung komplett umgeladen wird.

Abbildung 6: Versuchsaufbau zur Feldstrommessung



Die mit 2000 U/min rotierende Elektrode verfügt über unterschiedlich lange Arme und Plattenabstände. Diese Unsymmetrie ist bei der Strommessung im Feldkreis deutlich zu erkennen:

Abbildung 7: Feldstromverlauf

Der gemessene Feldstrom beträgt in der Spitze bis zu 1000 nA bei einem angelegten Feld von 15 kV/m. Sein Mittelwert beträgt 0 nA.

Der Influenzstrom ist also nicht nur auf die Sensorplatten beschränkt. Ein Influenzstrom fließt ebenso zwischen den felderzeugenden Platten. Da jeder Stromfluss bei Raumtemperatur verlustbehaftet ist, wird im Feldkreis Leistung umgesetzt.

Die hierzu notwendige Energie stammt entweder aus der Feldspannungsquelle oder aus dem Feld und führt daher in jedem Fall zu einer unerwünschten Beeinflussung des Feldkreises.

Die zur Deckung der Verluste benötigte Leistung kann man per Effektivstrom messen.

Zuerst wurde der Strom mit angelegtem Feld unterschiedlicher Spannung ohne Rotation gemessen. Es fließen konstant 52 nA_{eff} durch kapazitiv empfangene Netzwechselfeldspannungen, denn auch diese bewirken Wechselströme und Verluste im Feldkreis.

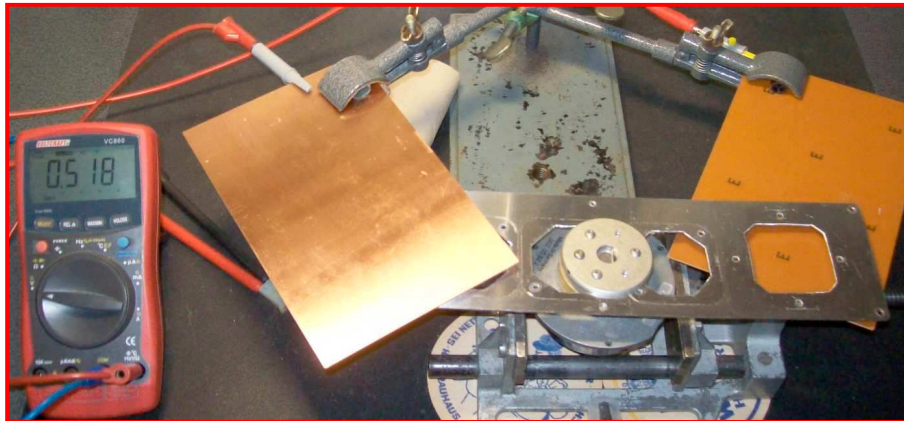
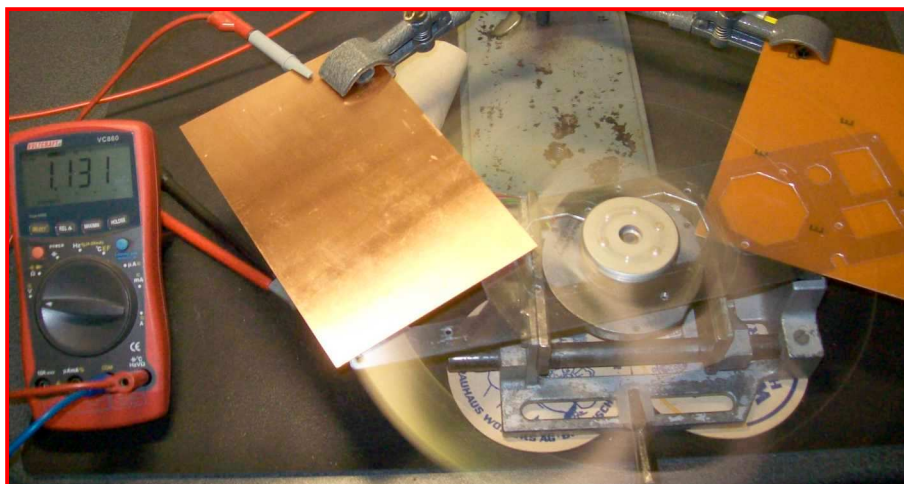


Abbildung 8: Feldstrom ohne Rotation



Sobald der Blechstreifen rotiert, steigt der Effektivstrom signifikant an

Abbildung 9: Feldstrom mit Rotation

Es ergaben sich folgende Messwerte:

Feldspannung	Strom ohne Rotation	Strom mit 2000 U/min	Nach Offset-Abzug
0 kV/m	52 nA _{eff}	52 nA _{eff}	0 nA _{eff}
7,5 kV/m	52 nA _{eff}	83 nA _{eff}	31 nA _{eff}
15 kV/m	52 nA _{eff}	113 nA _{eff}	61 nA _{eff}

Abbildung 10: Feldstrommessung

Die im Feldkreis fließenden Wechselströme bewirken messbare Verluste.

Rotationsvoltmeter vs. Integratormessmethode

Vom Prinzip her arbeiten Rotationsvoltmeter und das hier vorgestellte Messverfahren gleich. Beide messen den Influenzstrom zwischen Sensorelektroden.

Allerdings beeinflusst das Integratorverfahren das zu vermessende Feld nur dann mit Energieentnahme oder Energiezufuhr, wenn sich etwas an der vom Sensor erfassten Feldstärke ändert. Danach fließt kein weiterer Influenzstrom oder Feldstrom mehr und es treten auch keine Verluste mehr auf. Das Feld muss dann nur noch konstante Kraft aufbringen, um die verschobenen Ladungsträger in den Sensorplatten auf ihrer Position zu halten. Dies geschieht leistungslos.

Dagegen werden bei der Feldmühle zyklische Feldänderungen durch Rotationen oder Schwingungen der Sensorelektroden erzwungen, so dass diese Geräte zyklische Leitungsverluste bewirken und einen kontinuierlichen Energiefluss im Feld bzw. der Feldspannungsquelle bewirken.

Für die verbreitete Meinung, dass eine Feldmühle dem Feld keine Energie entnimmt, (bzw. zuführt) sieht der Verfasser keinen Anhaltspunkt.

Im Feldkreis fließt ein Wechselstrom. Dieser bewirkt Leitungsverluste. Die zu deren Deckung notwendige Energie stammt entweder aus der Feldspannungsquelle oder aus dem Feld und beeinflusst direkt die Feldspannung oder das elektrische Feld.

Fehlkonstruierte Influenzstrommessgeräte

(Influenz-)ströme misst man mit niederohmigen Amperemetern.

Ein hochohmiges stromdrosselndes Elektrometer verlängert die Zeit bis zum Influenzstromabklingen und vermindert gleichzeitig die Stromamplitude. Das raubt Empfindlichkeit, da kleinere Influenzströme über längere Zeit nachgewiesen werden müssen, was signaltechnisch ungünstig ist.

Bei Rotationsvoltmetern kann ein hochohmiger Eingang zusätzlich dazu führen, dass der Influenzstrom noch gar nicht abgeklungen ist, bevor die Sensormotor oder Schwinger die Stromrichtung des Influenzstromes schon wieder umkehrt. Dadurch erhält man eine unlineare Darstellung bei hohen Feldstärken.

Realisierung

Elektrodeninterface

Zur Abschätzung der zu messenden Stromstärken kann die Feldstärkeformel herangezogen werden:

Angenommene elektrische Feldstärke $E = 1000 \text{ [V/m]}$

Elektrische Feldkonstante $\epsilon_0 = 8,85 \text{ E-12 [As/Vm]}$

Dielektrizitätszahl für Luft $\epsilon_r = 1,00059$

Angenommene Fläche einer Sensor-Elektrode $A = 0,01 \text{ [m}^2\text{]}$

$$Q = E \epsilon_0 \epsilon_r A$$

Es fließt somit eine Ladungsmenge von rund 0.1 [nAs] pro 1000 [V/m] . Stromstärken im unteren Nanoamperebereich sind mit preiswerten Operationsverstärkern problemlos messbar. Um den zwischen den Anschlüssen „a“ und „b“ fließenden Strom – trotz erheblicher Brummeinkopplung – zu erfassen, wird für jede Achse X, Y und Z folgende stromgesteuerte Stromquelle verwendet:

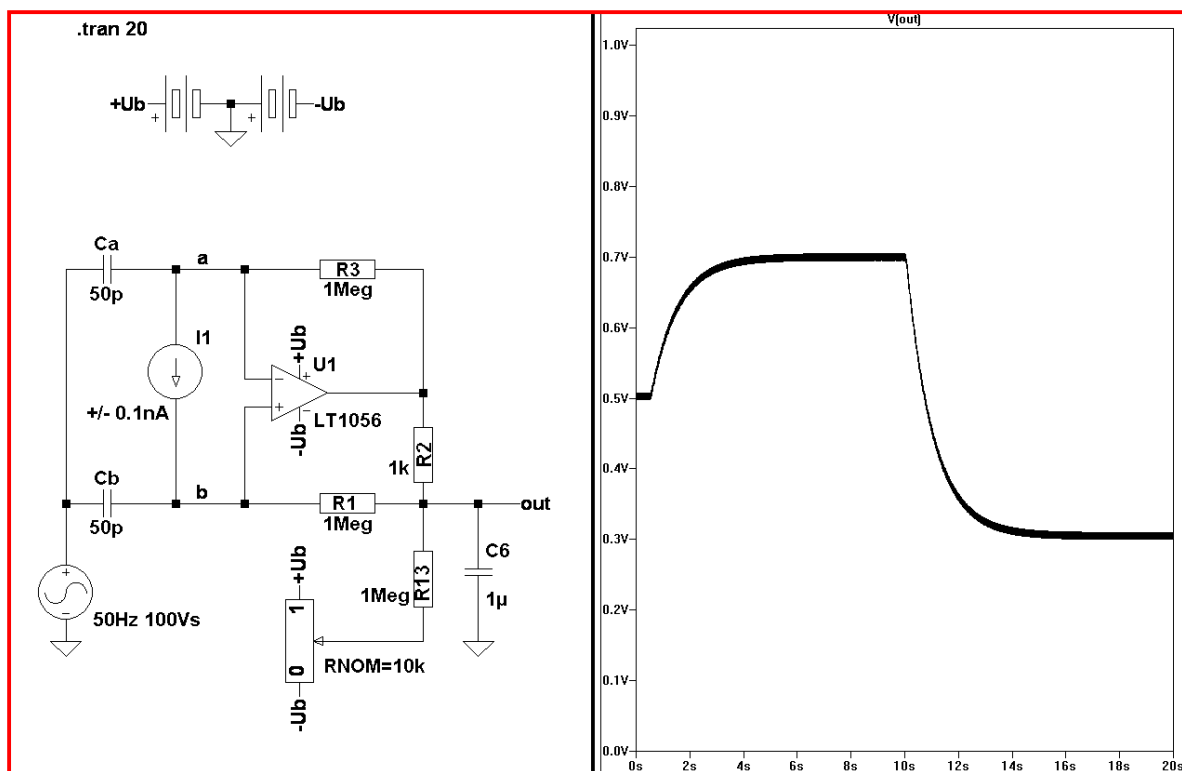


Abbildung 11: Stromgesteuerte Stromquelle

Sie erreicht eine gute Gleichtaktunterdrückung, eine Stromverstärkung von 2000, gestattet eine Entstörung mit C6 und kann den nachfolgenden Controllereingang ohne weitere Pegelschieber symmetrisch zu dessen +500mV Grundpegel ansteuern. Die bis zu 10mV große Offsetspannung des verwendeten LF347-Quad-OPVs und +500mV Grundpegel werden mit dem Trimmer kompensiert. Mit der gezeigten Dimensionierung sind Influenzströme aus Feldänderungen von bis zu 2kV/m innerhalb einer Sekunde ohne Übersteuerung erfassbar.

Gesamtschaltung

Die Elektronik besteht aus drei stromgesteuerten Stromquellen, einem Spannungswandler für den Quad-OPV und dem Controller mit Integrator- und USB-HiD-Software.

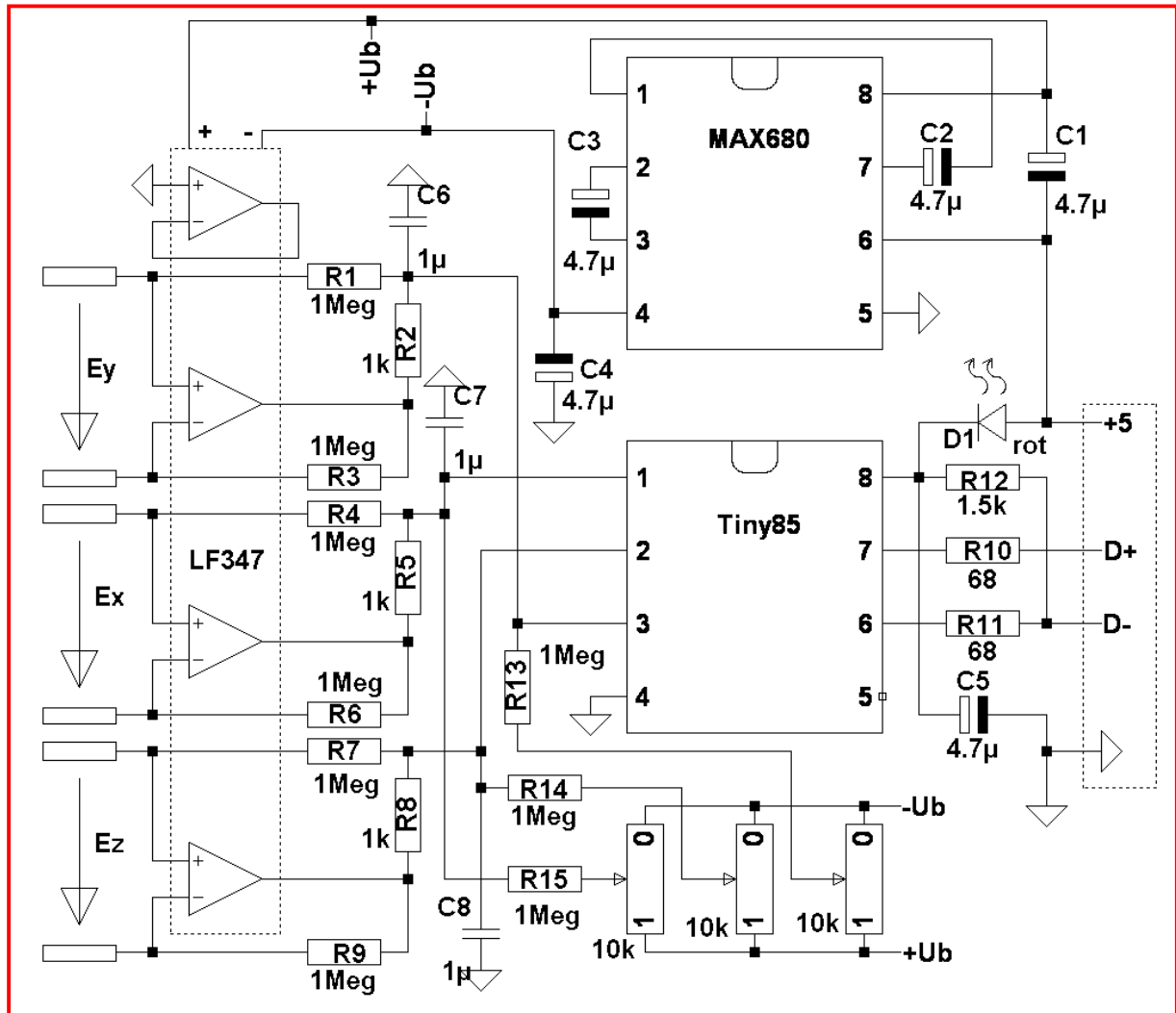


Abbildung 12: Gesamtschaltbild

Der Spannungswandler vervierfacht aus den +5V der USB-Versorgung eine symmetrische $\pm 10V$ Spannung. Diese OPV-Versorgungsspannung ist notwendig, um hohe Gleichtaktspannungen aus Wechselstromfeldern an den OPV-Eingängen subtrahieren zu können.

Zwei aus einseitig kaschierten Platinen hergestellten Elektroden einer Achse X, Y, Z stehen sich jeweils gegenüber und bilden das würfelförmige Gehäuse. Die Elektronik befindet sich im feldfreien Innenraum.

Die LED zeigt den Betrieb des Messgerätes an und adaptiert den Controller an die USB-Signalpegel.

Software

Die Software nutzt folgendes Tool:

```
/* Name: usbdv.c
 * Project: V-USB, virtual USB port for Atmel's(r) AVR(r) microcontrollers
 * Author: Christian Starkjohann
 * Creation Date: 2004-12-29
 * Tabsize: 4
 * Copyright: (c) 2005 by OBJECTIVE DEVELOPMENT Software GmbH
 * License: GNU GPL v2 (see License.txt), GNU GPL v3 or proprietary (CommercialLicense.txt)
 */
```

Sämtliche eigene Software befindet sich in einem Modul „e3d.c“ und ist selbsterklärend:

```
#include "usbdv/usbdv.h" // F_CPU in usbconfig.h
#include "usbdv/osccal.h"

#include <util/delay.h>
#include <avr/wdt.h>
#include <avr/interrupt.h>

#define FOREVER while(1)

/* Pin assignment:
 *
 * PB2 (7) = D+ (USB data line)
 * PB1 (6) = D- (USB data line)
 * PB0 (5) = Reserve (pull-up)
 * PB4 (3) = Y input (ADC2)
 * PB3 (2) = Z input (ADC3)
 * PB5 (1) = X input (ADC0)
 */

/* USB report descriptor */
PROGMEM const char usbHidReportDescriptor[USB_CFG_HID_REPORT_DESCRIPTOR_LENGTH] = {
    0x05, 0x01, // USAGE_PAGE (Generic Desktop)
    0x09, 0x06, // USAGE (Keyboard)
    0xa1, 0x01, // COLLECTION (Application)
    0x05, 0x07, // USAGE_PAGE (Keyboard)
    0x19, 0xe0, // USAGE_MINIMUM (Keyboard LeftControl)
    0x29, 0xe7, // USAGE_MAXIMUM (Keyboard Right GUI)
    0x15, 0x00, // LOGICAL_MINIMUM (0)
    0x25, 0x01, // LOGICAL_MAXIMUM (1)
    0x75, 0x01, // REPORT_SIZE (1)
    0x95, 0x08, // REPORT_COUNT (8)
    0x81, 0x02, // INPUT (Data,Var,Abs)
    0x95, 0x01, // REPORT_COUNT (1)
    0x75, 0x08, // REPORT_SIZE (8)
    0x25, 0x65, // LOGICAL_MAXIMUM (101)
    0x19, 0x00, // USAGE_MINIMUM (Reserved (no event indicated))
    0x29, 0x65, // USAGE_MAXIMUM (Keyboard Application)
    0x81, 0x00, // INPUT (Data,Ary,Abs)
    0xc0 // END_COLLECTION
};
```

```

#define NO_KEY          0
#define KEY_1           30
#define KEY_2           31
#define KEY_3           32
#define KEY_4           33
#define KEY_5           34
#define KEY_6           35
#define KEY_7           36
#define KEY_8           37
#define KEY_9           38
#define KEY_0           39
#define KEY_RETURN      40
#define KEY_TAB         43
#define ARROW_RIGHT     79
#define ARROW_LEFT     80
#define ARROW_DOWN     81
#define ARROW_UP       82
#define KEY_MINUS       0x2D
#define SHIFT_TAB       0xFE
#define LAST_KEY        0xFF

static const uchar keyReport[] = {
    KEY_0,
    KEY_1,
    KEY_2,
    KEY_3,
    KEY_4,
    KEY_5,
    KEY_6,
    KEY_7,
    KEY_8,
    KEY_9,
};

static uchar key[100];
static uchar key_ind;

static void store_key(uchar k)
{
    key[key_ind++] = k;
    key[key_ind++] = NO_KEY;
}

static void val_to_key(long val)                // long -> Tatstatur-Code
{
    if(val < 0) { store_key(KEY_MINUS); val = -val;}

    for(unsigned long div = 1000000;div;val %= div,div /= 10)
        store_key(keyReport[val / div]);
}

static struct {                                // 3D-Integrator und Entbrummer
    long x; long sum_x; long zero_x;
    long y; long sum_y; long zero_y;
    long z; long sum_z; long zero_z;
    long sum_temp; long temp;
    int cnt; unsigned char tara;
} integrator;

static void init_integrator(void)              //= "TARA"
{
    integrator.zero_x = integrator.zero_y = integrator.zero_z = 1024 / 2;
    integrator.sum_x = integrator.sum_y = integrator.sum_z = 0;
    integrator.sum_temp = integrator.temp = 0;
    integrator.x = integrator.y = integrator.z = 0;
    integrator.cnt = 0;
    integrator.tara = 5;
}

```

```

static void integrator_to_key(void)                // telegramm anfertigen
{
    key_ind = 0;
    val_to_key(integrator.x);
    store_key(KEY_TAB);

    val_to_key(integrator.y);
    store_key(KEY_TAB);

    val_to_key(integrator.z);
    store_key(KEY_TAB);

    val_to_key(integrator.temp);                  // ADC4 (Temperatur)

    store_key(SHIFT_TAB);
    store_key(SHIFT_TAB);
    store_key(SHIFT_TAB);
    store_key(KEY_RETURN);                        // nächste zeile ansteuern
    key[key_ind] = LAST_KEY;
    key_ind = 0;
}

static void sum_to_integrator(void)                // mathematik-routine
{
    integrator.sum_x /= integrator.cnt;           // brummen wegmitteln
    integrator.sum_y /= integrator.cnt;
    integrator.sum_z /= integrator.cnt;
    integrator.sum_temp /= integrator.cnt;

    if(integrator.sum_x > integrator.zero_x) integrator.zero_x++;
    else if(integrator.sum_x < integrator.zero_x) integrator.zero_x--;

    if(integrator.sum_y > integrator.zero_y) integrator.zero_y++;
    else if(integrator.sum_y < integrator.zero_y) integrator.zero_y--;

    if(integrator.sum_z > integrator.zero_z) integrator.zero_z++;
    else if(integrator.sum_z < integrator.zero_z) integrator.zero_z--;

    // die eigentliche integration machen
    integrator.x += (integrator.sum_x - integrator.zero_x);
    integrator.y += (integrator.sum_y - integrator.zero_y);
    integrator.z += (integrator.sum_z - integrator.zero_z);

    integrator.temp = integrator.sum_temp;        // temperatur einfach übernehmen

    integrator.cnt = 0;                           // brummglättung initialisieren
    integrator.sum_x = integrator.sum_y = integrator.sum_z = integrator.sum_temp = 0;
}

static uchar reportBuffer[2];                    // buffer for HID reports
static uchar idleRate;                          // in 4 ms units

static void buildReport(void)
{
    uchar c = key[key_ind];

    if(c == SHIFT_TAB) {
        reportBuffer[0] = _BV(1);                // left shift modifier
        reportBuffer[1] = KEY_TAB;
        key_ind++;
    }
    else if(c != LAST_KEY) {
        reportBuffer[0] = NO_KEY;
        reportBuffer[1] = c;
        key_ind++;
    }
    else reportBuffer[0] = reportBuffer[1] = NO_KEY;
}

```

```

#define MUX_DELAY      20                // us-wartezeit nach mux umschaltung

static void adc_to_sum(void)              // lies x,y,z und Temperatur
{
    DIDR0 = _BV(ADC0D) | _BV(ADC3D) | _BV(ADC2D);    // analog-eingänge
    ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS1) | _BV(ADPS0);    // 16 MHz / 128 = 125kHz

    ADMUX = _BV(REFS1);                    // ADC0
    _delay_us(MUX_DELAY);                  // s&h einpendeln lassen
    ADCSRA |= _BV(ADSC);                   // neue wandlung starten
    while(ADCSRA & _BV(ADSC));             // conversion complete?
    integrator.sum_x += ADC;

    ADMUX = _BV(REFS1) | _BV(MUX1);        // ADC2
    _delay_us(MUX_DELAY);                  // s&h einpendeln lassen
    ADCSRA |= _BV(ADSC);                   // neue wandlung starten
    while(ADCSRA & _BV(ADSC));             // conversion complete?
    integrator.sum_y += ADC;

    ADMUX = _BV(REFS1) | _BV(MUX1) | _BV(MUX0);    // ADC3
    _delay_us(MUX_DELAY);                  // s&h einpendeln lassen
    ADCSRA |= _BV(ADSC);                   // neue wandlung starten
    while(ADCSRA & _BV(ADSC));             // conversion complete?
    integrator.sum_z += ADC;

    ADMUX = _BV(REFS1) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1) | _BV(MUX0);    // ADC4 (Temperatur)
    _delay_us(MUX_DELAY);                  // s&h einpendeln lassen
    ADCSRA |= _BV(ADSC);                   // neue wandlung starten
    while(ADCSRA & _BV(ADSC));             // conversion complete?
    integrator.sum_temp += ADC;

    integrator.cnt++;                      // anzahl der messungen zählen
}

USB_PUBLIC void usbEventResetReady(void)    // calibrateOscillator
{
    calibrateOscillator();
}

USB_PUBLIC uchar usbFunctionSetup(unsigned char data[8])
{
    usbRequest_t *rq = (void *)data;
    usbMsgPtr = (void *)&reportBuffer;

    if((rq->bmRequestType & USBRQ_TYPE_MASK) == USBRQ_TYPE_CLASS){ /* class request type */
        /* wValue: ReportType (highbyte), ReportID (lowbyte) */
        /* we only have one report type, so don't look at wValue */
        if(rq->bRequest == USBRQ_HID_GET_REPORT){
            buildReport();
            return sizeof(reportBuffer);
        } else if(rq->bRequest == USBRQ_HID_GET_IDLE){
            usbMsgPtr = &idleRate;
            return 1;
        } else if(rq->bRequest == USBRQ_HID_SET_IDLE){
            idleRate = rq->wValue.bytes[1];
        }
    }
    // else no vendor specific requests implemented
    return 0;
}

```

```

int main(void)
{
    uint16_t i;

    wdt_enable(WDTO_2S);

    usbInit();

    usbDeviceDisconnect();           // enforce re-enumeration
    for(i = 50; i--; ) {             // wait 500 ms
        wdt_reset();                 // keep the watchdog happy
        _delay_ms(10);
    }
    usbDeviceConnect();

    PORTB |= _BV(PB0);               // pull-up

    sei(); // Enable interrupts after re-enumeration

    key[key_ind = 0] = LAST_KEY;
    init_integrator();

    i = 10000;                        // alle 10 Sekunden ein Telegramm
    FOREVER {                          // main event loop
        _delay_ms(1);                 // 1000 Msg./s
        wdt_reset();
        usbPoll();
        wdt_reset();

        adc_to_sum();                 // brummen mitteln

        if(!--i) {                   // alle 10 sekunden Telegramm schnüren
            if(!integrator.tara) integrator_to_key();
            else integrator.tara--; // einlaufzeit
            i = 10000;                // nächste 10 Sekunden starten
            continue;
        }
        if(!(i % 100)) {              // alle 100ms entbrummtes integrieren
            sum_to_integrator();
            continue;
        }
        if((i + 50) % 100) continue; // alle 100ms taste drücken

        if((key[key_ind] != LAST_KEY)&&(usbInterruptIsReady())) {
            buildReport();             // nächstes zeichen übertragen
            usbSetInterrupt((void *)&reportBuffer, sizeof(reportBuffer));
        }
    }
    return 0;
}
/* ENDE */

```

Abbildung 13: Software-Listing e3d.c

Anmerkung: die hier abgedruckte Software ist lauffähig. Sie hat aber ein paar kleinere Mängel, deren Behebung nicht in die Doku eingepflegt werden, da hier nur das Prinzip gezeigt wird. Bei Interesse bitte den Verfasser nach einer aktuelle Software-Version bzw. Hexdatei fragen.

Aufbau

Zur Zeit befindet sich die Schaltung noch auf einem Steckbrett.

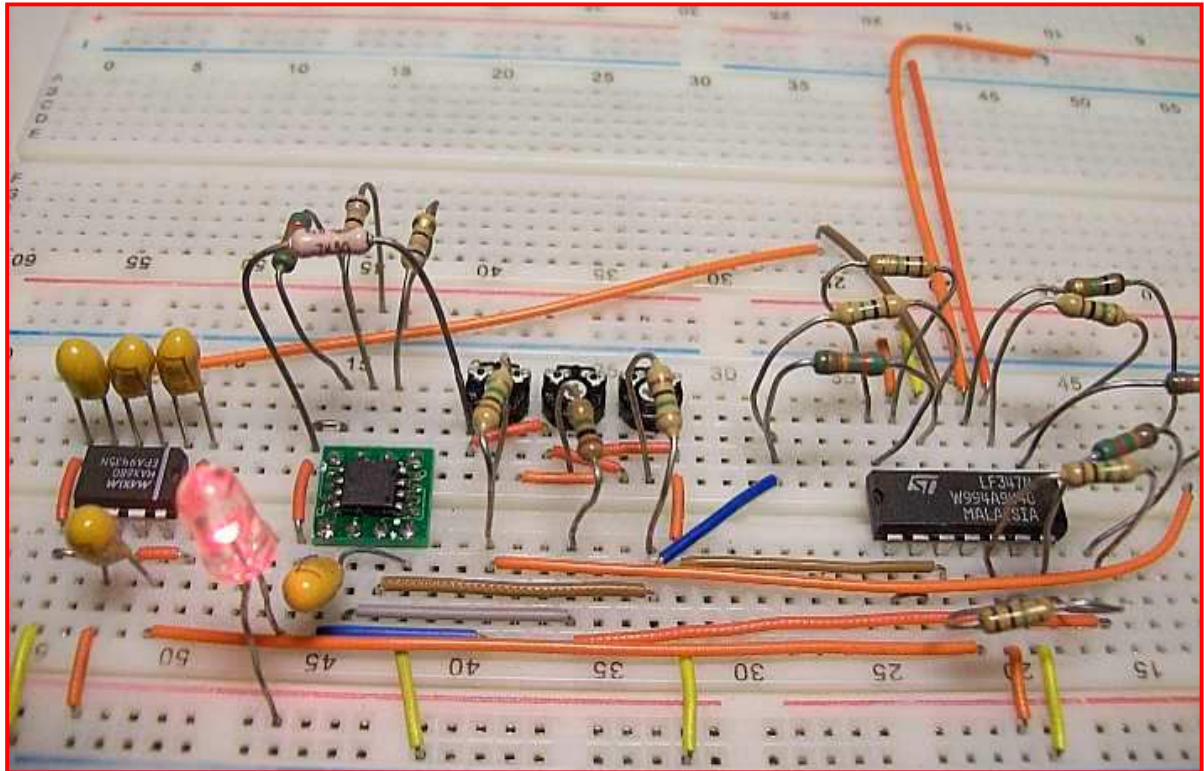


Abbildung 14: Steckbrettaufbau

Messungen

Für nachfolgende Messungen wurde ein Android-Tablet verwendet, auf dem eine „Kingsoft“-Tabellenkalkulation die Datenerfassung durchführt. Das Tablet versorgt die Sensor-Hardware und empfängt alle 10 Sekunden ein Tastaturzeichen-Telegramm mit den drei Messkanälen und der Temperatur.

Nach der Datenerfassung wurde das Tablet mit einem PC verbunden und die Kingsoft-Daten per OpenOffice grafisch ausgewertet.

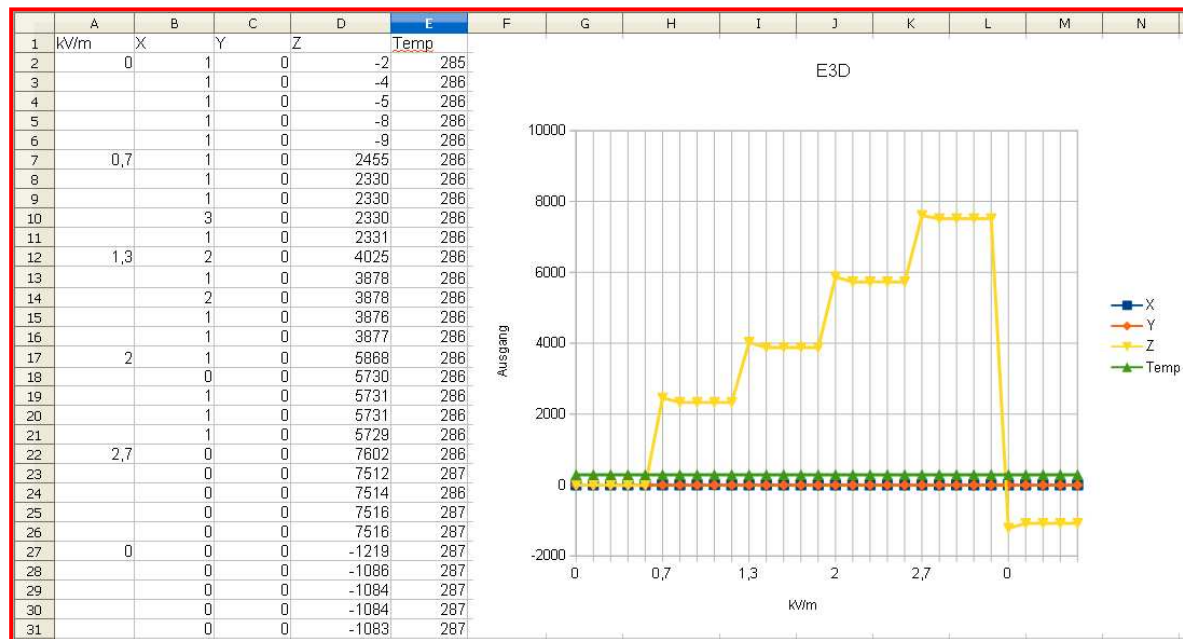


Abbildung 15: Messdaten

An Kanal X und Y wurden keine Elektroden angeschlossen. Die „Temp“-Spalte zeigt die Temperatur der Elektronik. Die Sensor-Elektroden des Kanals Z wurden variablen Feldern ausgesetzt.

Leider sind die Feldplatten den Sensorelektroden – zur Umgehung von Hochspannung - relativ nahe, so dass eine kapazitive Kopplung sichtbar wird. Aus gleichem Grund wurde der Nullwert beim schlagartigen Rücksprung von 2,7 kV/m auf 0 kV/m nicht sonderlich gut wiedergefunden. Derartige Effekte verschwinden bei realen Messungen.

Insgesamt sieht man die Stabilität der drei Integratoren. Die Messgrenze liegt bei 30 V/m! Das Gerät reagiert mit heftigen Anzeigen, wenn sich jemand im Raum bewegt.

Anwendungen

- Bewegungsmelder
- Robotik
- Halbleiterproduktion
- Berührungsfreie Gleichspannungsüberwachung
- Feldgeometrie-Untersuchung