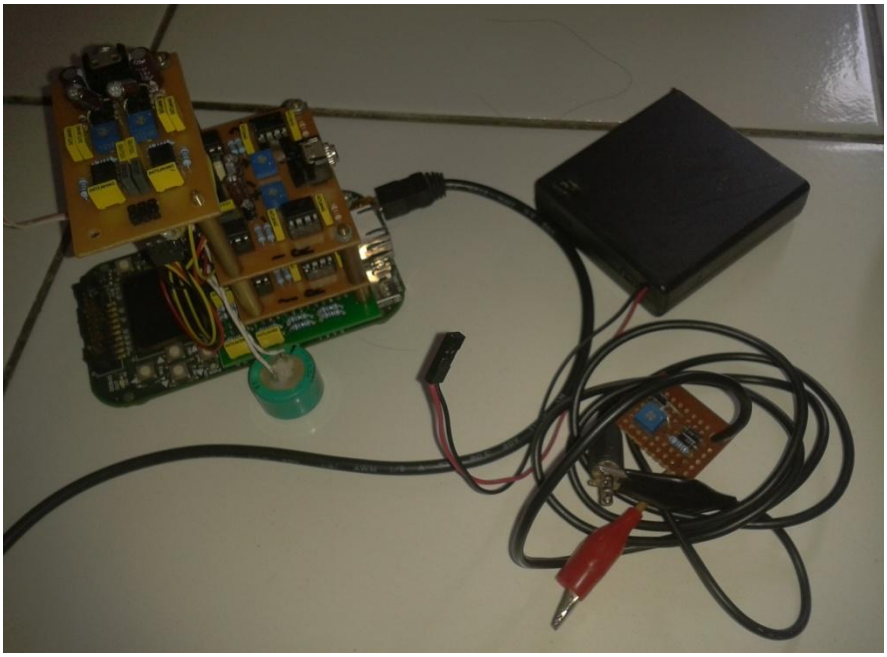
LAMPIRAN A

FOTO HASIL PERANCANGAN ACTIVE NOISE REDUCTION

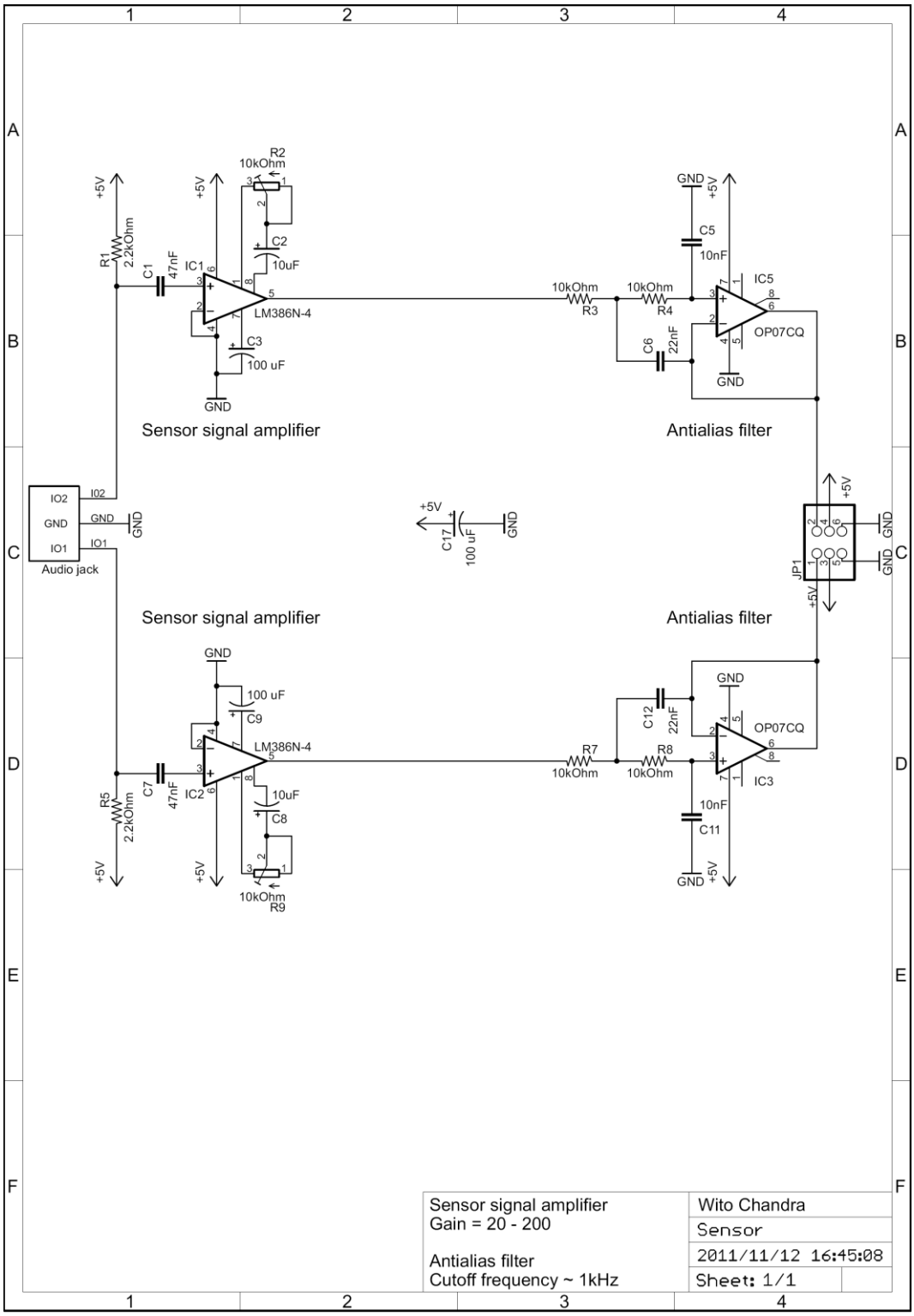
(Headphone)

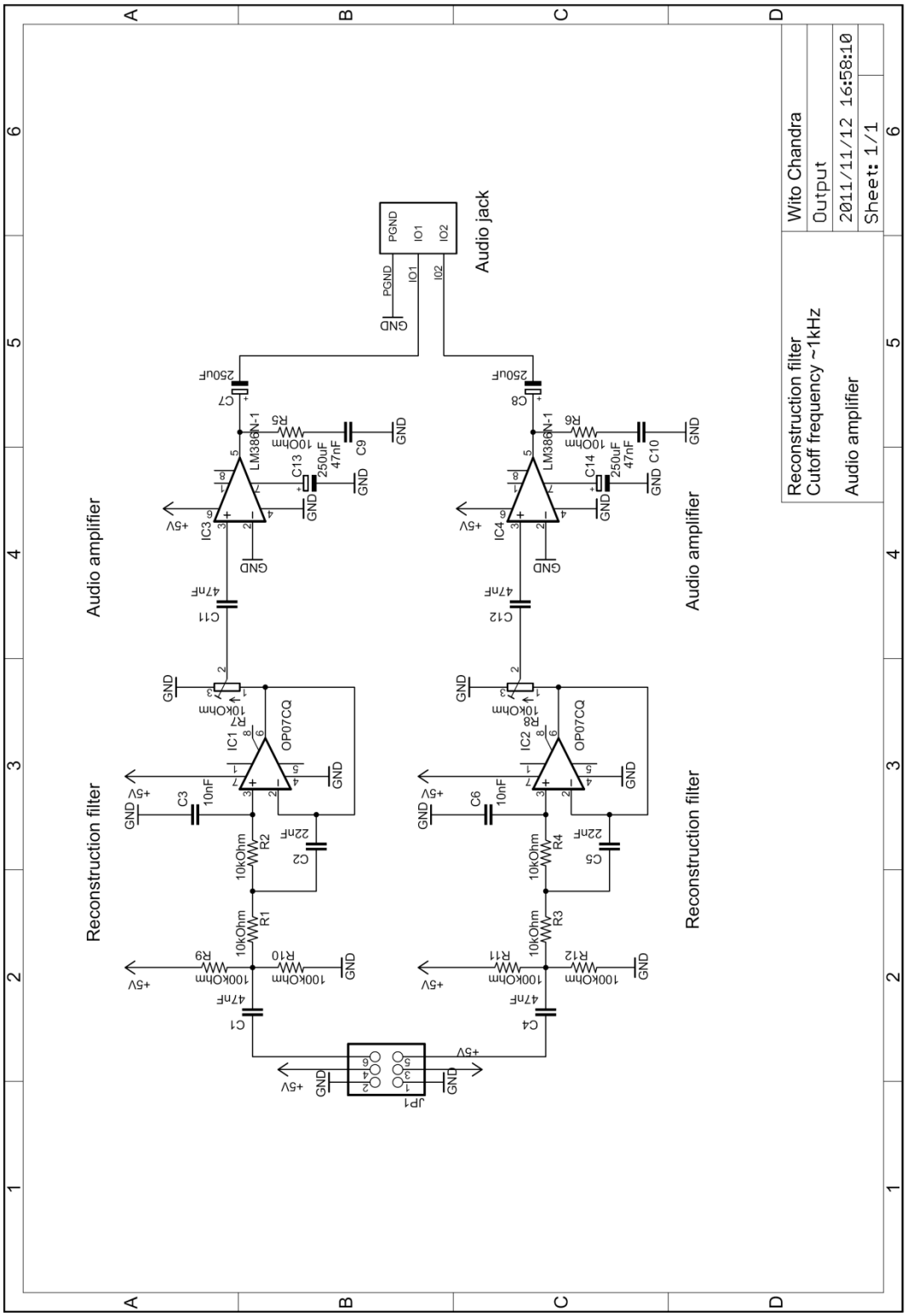


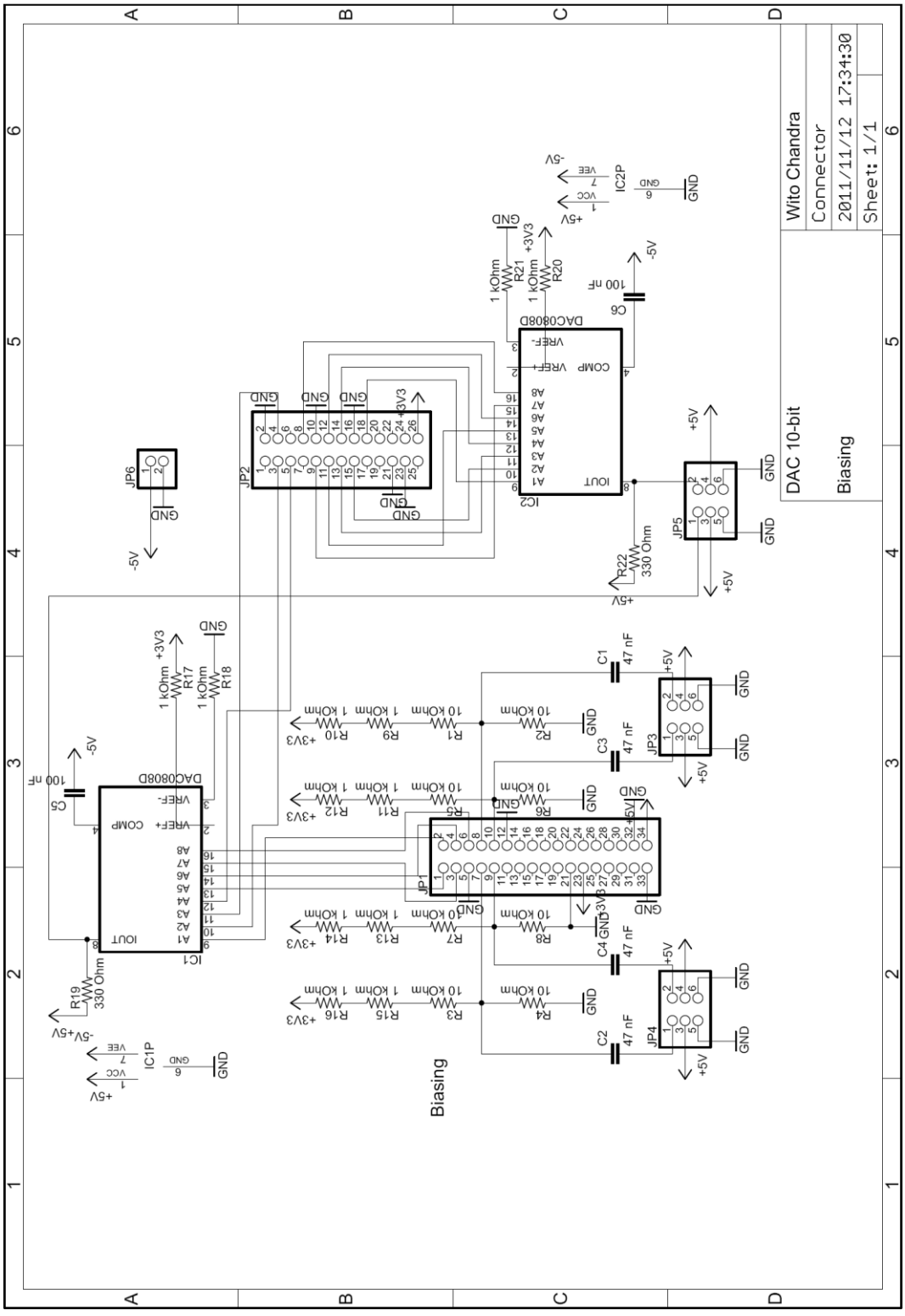
(Rangkaian *active noise reduction)*

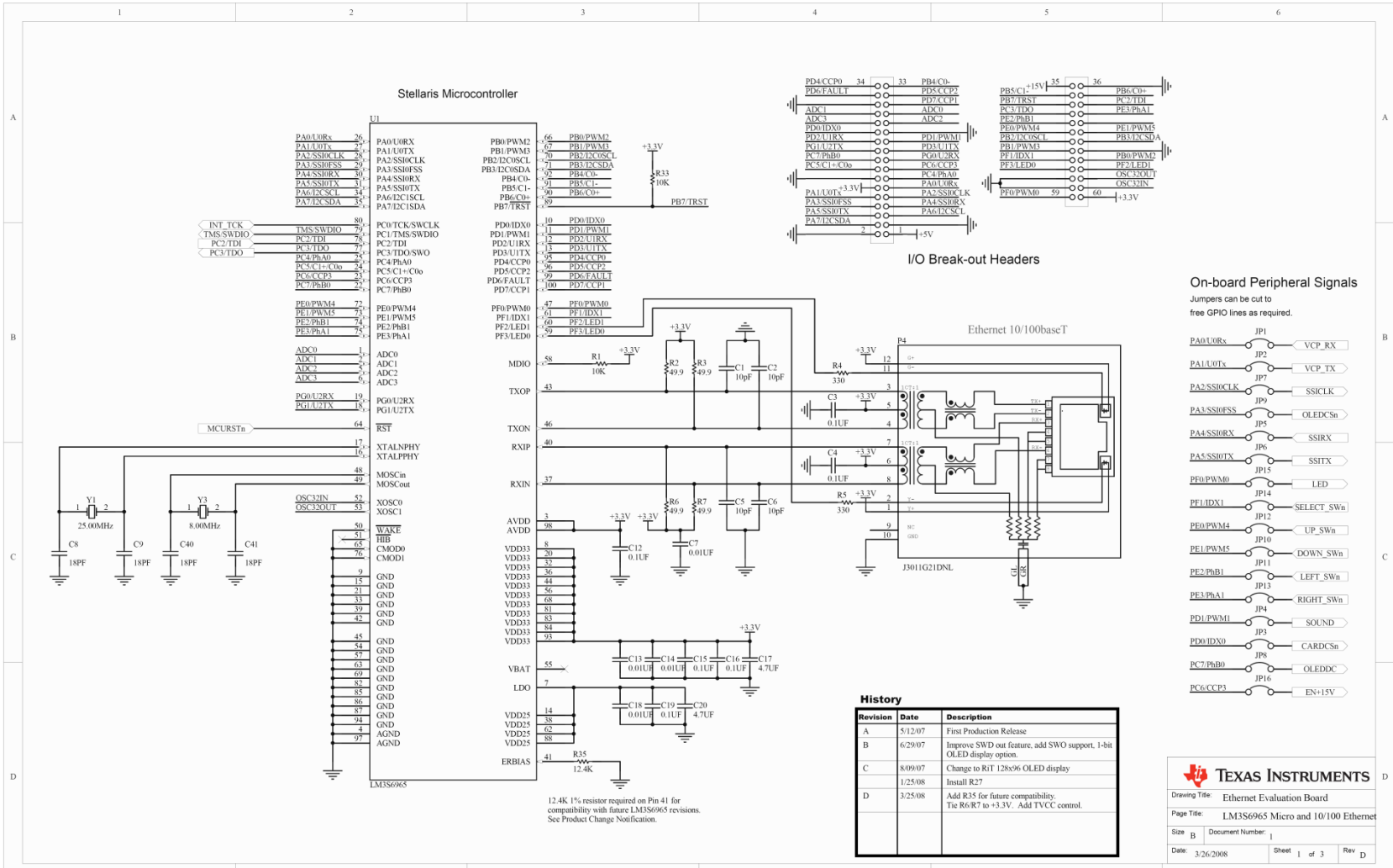
LAMPIRAN B

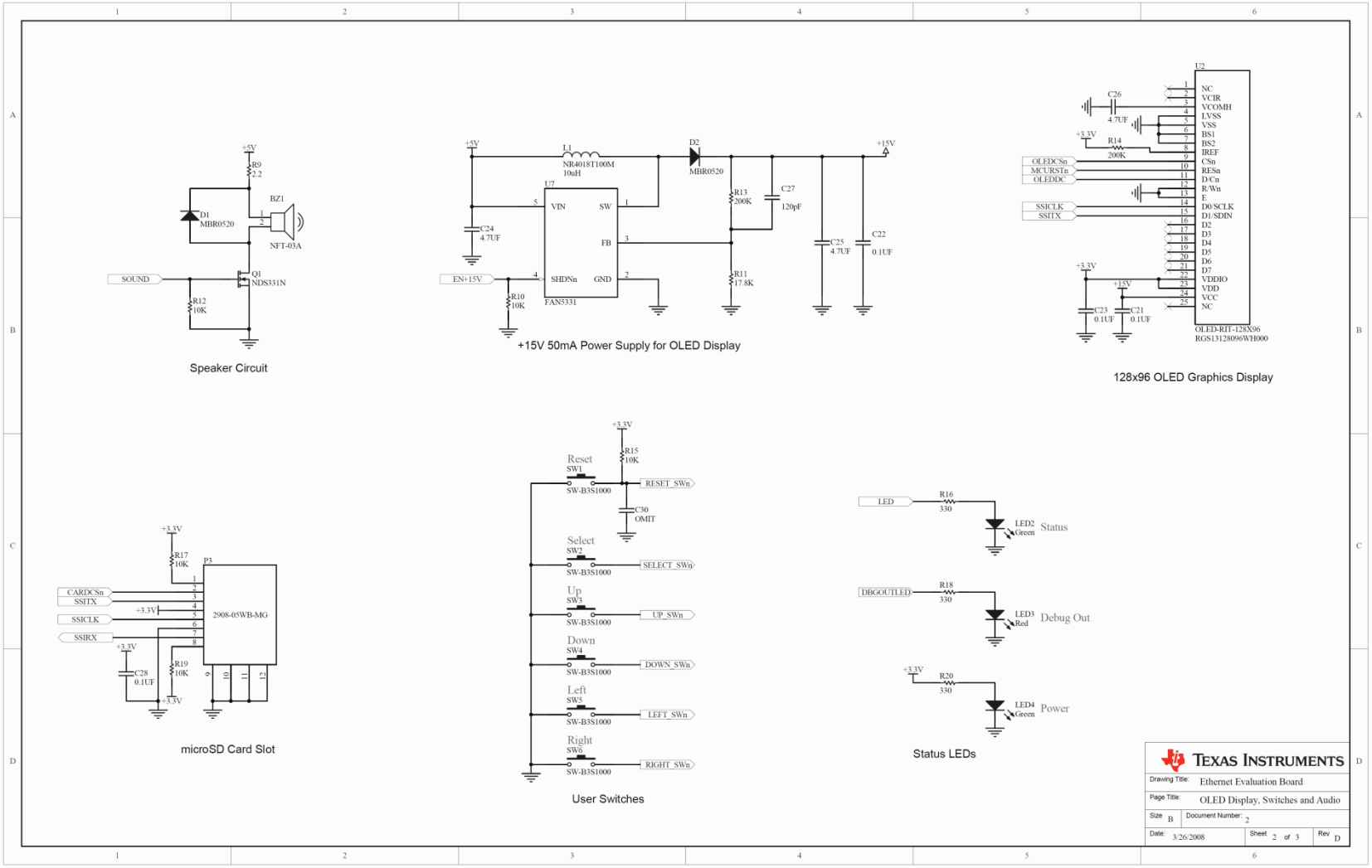
RANGKAIAN ACTIVE NOISE REDUCTION

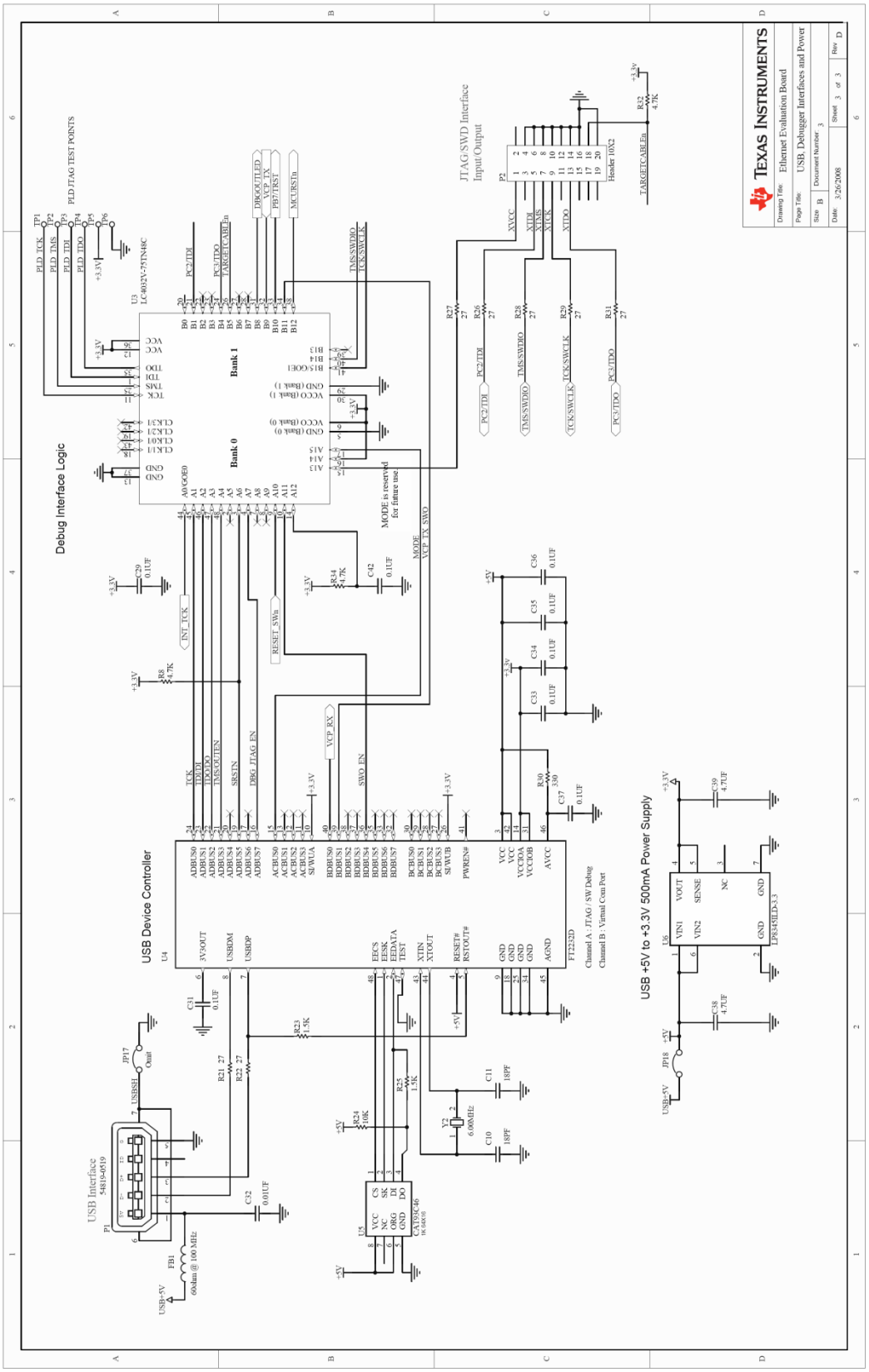



Reconstruction filter
Cutoff frequency ~1kHz
Audio amplifier
Wito Chandra
Output
2011/11/12 16:58:10
Sheet: 1/1
Audio jack
PGND
IO1
IO2
JP1
Audio amplifier
Reconstruction filter
LM386N-1
OP07CQ
10kOhm
100kOhm
10nF
22nF
47nF
250uF
250uF
10Ohm
+5V
GND


DAC 10-bit
Wito Chandra
Connector
2011/11/12 17:34:30
Sheet: 1/1
Biasing
DAC0808D
IC1
IC2
IC1P
IC2P
JP1
JP2
JP3
JP4
JP5
JP6
C1
C2
C3
C4
C5
C6
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
10 kOhm
1 kOhm
330 Ohm
47 nF
100 nF
+5V
-5V
+3V3
GND


Stellaris Microcontroller
U1
LM3S6965
I/O Break-out Headers
Ethernet 10/100baseT
J3011G21DNL
On-board Peripheral Signals
Jumpers can be cut to
free GPIO lines as required.
VCP_RX
VCP_TX
SSICLK
OLEDCSn
SSIRX
SSITX
LED
SELECT_SWn
UP_SWn
DOWN_SWn
LEFT_SWn
RIGHT_SWn
SOUND
CARDCSn
OLEDDC
EN+15V
12.4K 1% resistor required on Pin 41 for
compatibility with future LM3S6965 revisions.
See Product Change Notification.
History
Revision
Date
Description
A
5/12/07
First Production Release
B
6/29/07
Improve SWD out feature, add SWO support, 1-bit OLED display option.
C
8/09/07
Change to RiT 128x96 OLED display
1/25/08
Install R27
D
3/25/08
Add R35 for future compatibility. Tie R6/R7 to +3.3V. Add TVCC control.
Texas Instruments
Drawing Title: Ethernet Evaluation Board
Page Title: LM3S6965 Micro and 10/100 Ethernet
Size B
Document Number: 1
Date: 3/26/2008
Sheet 1 of 3
Rev D


Speaker Circuit
+15V 50mA Power Supply for OLED Display
128x96 OLED Graphics Display
microSD Card Slot
User Switches
Status LEDs
Texas Instruments
Drawing Title: Ethernet Evaluation Board
Page Title: OLED Display, Switches and Audio
Size: B
Document Number: 2
Date: 3/26/2008
Sheet 2 of 3
Rev D


TEXAS INSTRUMENTS
Drawing Title: Ethernet Evaluation Board
Page Title: USB, Debugger Interfaces and Power
Size: B
Document Number: 3
Date: 3/26/2008
Sheet: 3 of 3
Rev: D
Debug Interface Logic
JTAG/SWD Interface
Input/Output
USB Device Controller
USB Interface
54819-0619
USB +5V to +3.3V 500mA Power Supply
PLD JTAG TEST POINTS
Bank 0
Bank 1
MODE is reserved for future use
Channel A : JTAG / SW Debug
Channel B : Virtual Com Port
FT2232D
LP3845ILD-3.3
LC4032V-75TN48C
Header 10X2

LAMPIRAN C

LISTING PROGRAM

main.c

```
#include "inc/lm3s6965.h"
#include "inc/hw_types.h"
#include "drivers/rit128x96x4.h"
#include "driverlib/sysctl.h"
#include "utilities.h"
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define ORDER 32

char buffer[32];
long adcBuffer[4];
long i = 0;
long pointer = 0;
float leftInput[ORDER];
float leftError;
float rightInput[ORDER];
float rightError;
float leftWeight[ORDER];
float rightWeight[ORDER];
float leftOutput = 0;
float rightOutput = 0;
long j = 0;
float desire[8] = {0, .175, .25, .175, 0, -.175, -.25, -.175};

int main(){
    initialize();
    setStatusLed(true);
    for(i = 0; i < ORDER; i++){
        leftInput[i] = 0;
        leftWeight[i] = 0;
        rightInput[i] = 0;
        rightWeight[i] = 0;
    }
    while(getKeypad(0) == KEY_TIMEOUT){
        for(i = ORDER - 1; i > 0; i --){
            leftInput[i] = leftInput[i - 1];
            rightInput[i] = rightInput[i - 1];
        }
        getAdc(adcBuffer);
        leftInput[0] = (((float)adcBuffer[0]/512) - 1);
        leftError = desire[j] -
```

```
                    ((float)adcBuffer[2]/512)  + 1;
        rightInput[0] = ((float)adcBuffer[1]/512) - 1;
        rightError = desire[j] - ((float)adcBuffer[3]/512) +
                             1;
        for(i = 0; i < ORDER; i ++){
            leftWeight[i] += (.1F*leftInput[i]*leftError);
            rightWeight[i] += (.1F*rightInput[i]*rightError);
        }
        leftOutput = 0;
        rightOutput = 0;
        for(i = 0; i < ORDER; i ++){
            leftOutput += leftInput[i]*leftWeight[i];
            rightOutput += rightInput[i]*rightWeight[i];
        }
        j++;
        if(j == 8){
            j = 0;
        }
        leftOutput += desire[j];
        rightOutput += desire[j];
        if(leftOutput > 1){
            leftOutput = 1;
        } else if(leftOutput < -1){
            leftOutput = -1;
        }
        if(rightOutput > 1){
            rightOutput = 1;
        } else if(rightOutput < -1){
            rightOutput = -1;
        }
        setDac1((rightOutput + 1) * 127);
        setDac2((leftOutput + 1) * 127);
        delayUs(30); //orde 32
        //delayUs(345); //orde 8
        //delayUs(400); //orde 4
    }
    i = 0;
    while(i < ORDER){
        RIT128x96x4Clear();
        for(i = pointer; (i < ORDER) && (i < pointer + 9); i++){
            sprintf(buffer, "W%d = %d", i,
                 (int)(leftWeight[i]*1000));
            RIT128x96x4StringDraw(buffer, 0, 10 * (i - pointer),
                    8, STYLE_NORMAL);
        }
        while(getKeypad(0) == KEY_TIMEOUT);
        pointer += 9;
    }
}
```

utilities.h

```
#ifndef __UTILITIES_H__
#define __UTILITIES_H__

#include "inc/hw_types.h"

#define KEY_SELECT     0x00000000
#define KEY_TIMEOUT    0xFFFFFFFF

void initialize(void);
//inisialisasi mikrokontroler LM3S6965

void setStatusLed(tBoolean state);
//mengatur keadaan led status

int getKeypad(int timeoutms);
//me-return tombol yang ditekan selama timeoutms

void delayMs(int timems);
//delay selama timems milisekon

void delayUs(int timeus);
//delay selama timeus mikrosekon

void getAdc(long * values);
//membaca nilai – nilai adc

void setDac1(int data);
//mengatur nilai DAC1

void setDac2(int data);
//mengatur nilai DAC1

#endif
```

utilities.c

```
#include "utilities.h"
#include "inc/hw_pwm.h"
#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "inc/lm3s6965.h"
#include "driverlib/pwm.h"
#include "driverlib/gpio.h"
#include "driverlib/sysctl.h"
#include "driverlib/systick.h"
#include "drivers/rit128x96x4.h"
#include "driverlib/adc.h"
#include "driverlib/uart.h"
#include <stdlib.h>

void initialize(void){
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL |
            SYSCTL_XTAL_8MHZ | SYSCTL_OSC_MAIN);
    SYSCTL_RCGC0_R |= 0x00010300;

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOG);

    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1
            | GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 |
            GPIO_PIN_6);
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_4 | GPIO_PIN_5
            | GPIO_PIN_6 | GPIO_PIN_7);
    GPIOPinTypeGPIOOutput(GPIO_PORTE_BASE, GPIO_PIN_0 | GPIO_PIN_1
            | GPIO_PIN_2 | GPIO_PIN_3);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_0);
    GPIOPinTypeGPIOOutput(GPIO_PORTG_BASE, GPIO_PIN_0);
    GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_1);
    GPIOPadConfigSet(GPIO_PORTF_BASE, GPIO_PIN_1,
            GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD_WPU);

    RIT128x96x4Init(1000000);

    ADCSequenceConfigure(ADC0_BASE, 1, ADC_TRIGGER_PROCESSOR, 0);
    ADCSequenceStepConfigure(ADC0_BASE, 1, 0,ADC_CTL_CH0);
    ADCSequenceStepConfigure(ADC0_BASE, 1, 1,ADC_CTL_CH1);
    ADCSequenceStepConfigure(ADC0_BASE, 1, 2,ADC_CTL_CH2);
    ADCSequenceStepConfigure(ADC0_BASE, 1, 3,ADC_CTL_IE |
            ADC_CTL_END | ADC_CTL_CH3);
    ADCSequenceEnable(ADC0_BASE, 1);
```

```
    ADCReferenceSet(ADC0_BASE, ADC_REF_INT);
}

void setStatusLed(tBoolean state){
    if(state){
        GPIO_PORTF_DATA_R |= 0x01;
    } else {
        GPIO_PORTF_DATA_R &= 0xFE;
    }
}

int getKeypad(int timeoutms){
    int i = 0;
    if((GPIO_PORTF_DATA_R & 0x02) == 0x00){
        while((GPIO_PORTF_DATA_R & 0x02) == 0x00);
        return KEY_SELECT;
    }
    for(i = 0; i < timeoutms; i++){
        if((GPIO_PORTF_DATA_R & 0x02) == 0x00){
            while((GPIO_PORTF_DATA_R & 0x02) == 0x00);
            return KEY_SELECT;
        }
        delayMs(1);
    }
    return KEY_TIMEOUT;
}

void delayMs(int timems){
    SysCtlDelay(timems * 50000 / 3);
}

void delayUs(int timeus){
    SysCtlDelay(timeus * 50 / 3);
}

void getAdc(long * values){
    ADCIntClear(ADC0_BASE, 1);
    ADCProcessorTrigger(ADC0_BASE, 1);
    while(!ADCIntStatus(ADC0_BASE, 1, false)){};
    ADCSequenceDataGet(ADC0_BASE, 1, values);
}

void setDac1(int data){
    char dataB = 0;
    char dataD = 0;
    char dataG = 0;
    int i = 0;
    for(i = 0; i < 4; i++){
        dataD = dataD << 1;
```

```
        dataD |= (data >> i) & 0x01;
    }
    dataD = (dataD << 4) & 0xF0;
    for(i = 5; i < 8; i++){
        dataB = dataB << 1;
        dataB |= (data >> i) & 0x01;
    }
    dataB = (dataB << 4) & 0x70;
    dataG = (data >> 4) & 0x01;
    GPIO_PORTB_DATA_R &= 0x0F;
    GPIO_PORTB_DATA_R |= dataB;
    GPIO_PORTD_DATA_R &= 0x0F;
    GPIO_PORTD_DATA_R |= dataD;
    GPIO_PORTG_DATA_R &= 0xFE;
    GPIO_PORTG_DATA_R |= dataG;
}

void setDac2(int data){
    char dataB = 0;
    char dataE = 0;
    int i = 0;
    for(i = 0; i < 4; i++){
        dataE = dataE << 1;
        dataE |= (data >> i) & 0x01;
    }
    dataE = dataE & 0x0F;
    for(i = 4; i < 8; i++){
        dataB = dataB << 1;
        dataB |= (data >> i) & 0x01;
    }
    dataB = dataB & 0x0F;
    GPIO_PORTB_DATA_R &= 0xF0;
    GPIO_PORTB_DATA_R |= dataB;
    GPIO_PORTE_DATA_R &= 0xF0;
    GPIO_PORTE_DATA_R |= dataE;
}
```

LAMPIRAN D

DATASHEET

C H A P T E R 1

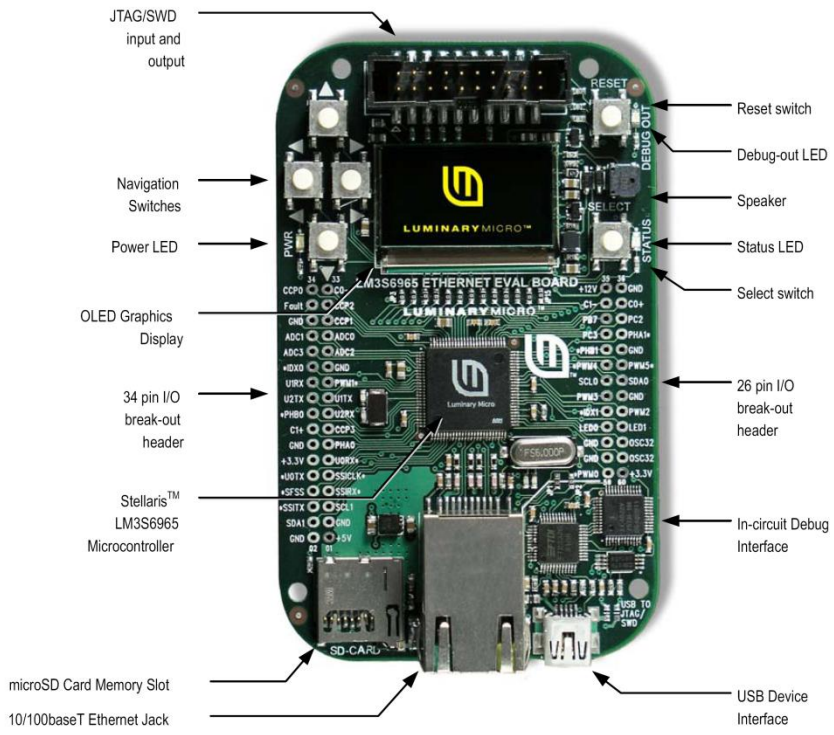
Stellaris® LM3S6965 Evaluation Board

The Stellaris® LM3S6965 Evaluation Board is a compact and versatile evaluation platform for the Stellaris LM3S6965 ARM® Cortex™-M3-based microcontroller. The evaluation kit uses the LM3S6965 microcontroller's fully integrated 10/100 Ethernet controller to demonstrate an embedded web server.

You can use the board either as an evaluation platform or as a low-cost in-circuit debug interface (ICDI). In debug interface mode, the on-board microcontroller is bypassed, allowing programming or debugging of an external target. The kit is also compatible with high-performance external JTAG debuggers.

This evaluation kit enables quick evaluation, prototype development, and creation of application-specific designs for Ethernet networks. The kit also includes extensive source-code examples, allowing you to start building C code applications quickly.

Figure 1-1. Stellaris LM3S6965 Evaluation Board Layout



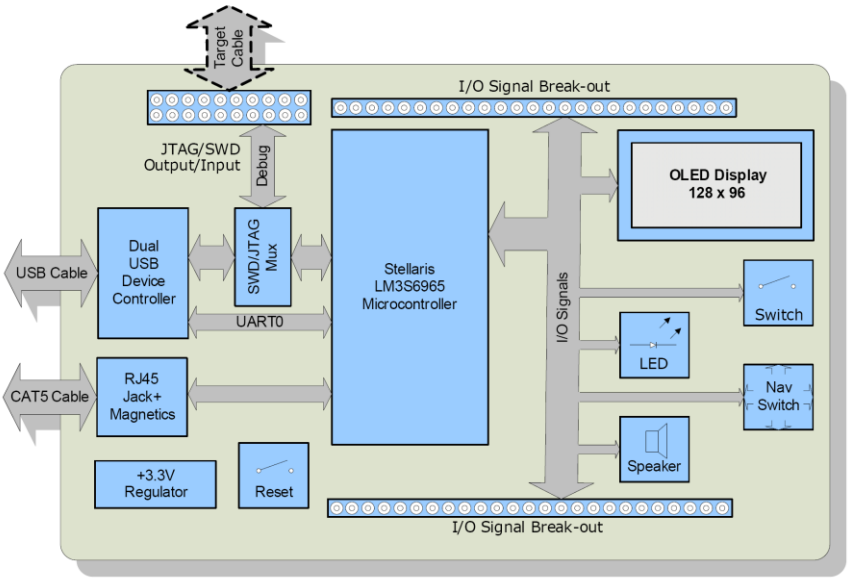
Features

The Stellaris LM3S6965 Evaluation Board includes the following features:

- Stellaris LM3S6965 microcontroller with fully-integrated 10/100 embedded Ethernet controller
- Simple setup; USB cable provides serial communication, debugging, and power
- OLED graphics display with 128 x 96 pixel resolution
- User LED, navigation switches, and select pushbuttons
- Magnetic speaker
- LM3S6965 I/O available on labeled break-out pads
- Standard ARM® 20-pin JTAG debug connector with input and output modes
- USB interface for debugging and power supply
- MicroSD card slot

Block Diagram

Figure 1-2. LM3S6965 Evaluation Board Block Diagram



Evaluation Kit Contents

The evaluation kit contains everything needed to develop and run applications for Stellaris microcontrollers including:

- LM3S6965 Evaluation Board (EVB)
- USB cable
- 20-pin JTAG/SWD target cable
- CD containing:
  - A supported version of one of the following (including a toolchain-specific Quickstart guide):
    - Keil™ RealView® Microcontroller Development Kit (MDK-ARM)
    - IAR Embedded Workbench
    - Code Sourcery GCC development tools
    - Code Red Technologies development tools
    - Texas Instruments' Code Composer Studio™ IDE
  - Complete documentation
  - Quickstart application source code
  - Stellaris® Firmware Development Package with example source code

Evaluation Board Specifications

- Board supply voltage: 4.37–5.25 Vdc from USB connector
- Board supply current: 250 mA typ (fully active, CPU at 50 MHz)
- Break-out power output: 3.3 Vdc (60 mA max), 15 Vdc (15 mA max)
- Dimensions: 4.0" x 2.45" x 0.7" (LxWxH)
- RoHS status: Compliant

Features of the LM3S6965 Microcontroller

- 32-bit RISC performance using ARM® Cortex™-M3 v7M architecture
  - 50-MHz operation
  - Hardware-division and single-cycle-multiplication
  - Integrated Nested Vectored Interrupt Controller (NVIC)
  - 42 interrupt channels with eight priority levels
- 256 KB single-cycle Flash
- 64 KB single-cycle SRAM
- Four general-purpose 32-bit timers
- Integrated Ethernet MAC and PHY
- Three fully programmable 16C550-type UARTs
- Four 10-bit channels (inputs) when used as single-ended inputs

- Two independent integrated analog comparators
- Two I^2C modules
- Three PWM generator blocks
  - One 16-bit counter
  - Two comparators
  - Produces two independent PWM signals
  - One dead-band generator
- Two QEI modules with position integrator for tracking encoder position
- 0 to 42 GPIOs, depending on user configuration
- On-chip low drop-out (LDO) voltage regulator

National Semiconductor

August 2000

LM386
Low Voltage Audio Power Amplifier

General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value from 20 to 200.

The inputs are ground referenced while the output automatically biases to one-half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

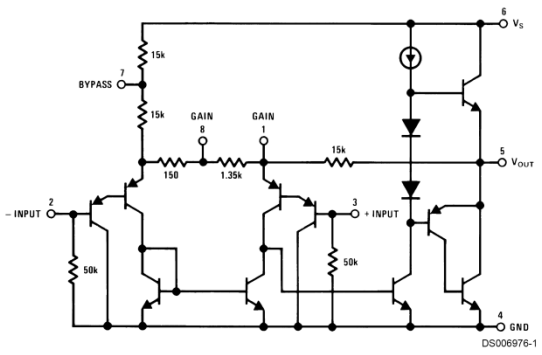
Features

- Battery operation
- Minimum external parts
- Wide supply voltage range: 4V–12V or 5V–18V
- Low quiescent current drain: 4mA
- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion: 0.2% (A_V = 20, V_S = 6V, R_L = 8Ω, P_O = 125mW, f = 1kHz)
- Available in 8 pin MSOP package

Applications

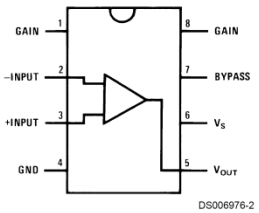
- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

Equivalent Schematic and Connection Diagrams



DS006976-1

Small Outline,
Molded Mini Small Outline,
and Dual-In-Line Packages



DS006976-2

Top View
Order Number LM386M-1, LM386MM-1, LM386N-1, LM386N-3 or LM386N-4
See NS Package Number M08A, MUA08A or N08E

Ultralow Offset Voltage Operational Amplifier

OP07

FEATURES

Low V_{OS}: 75 µV maximum
Low V_{OS} drift: 1.3 µV/°C maximum
Ultrastable vs. time: 1.5 µV per month maximum
Low noise: 0.6 µV p-p maximum
Wide input voltage range: ±14 V typical
Wide supply voltage range: 3 V to 18 V
125°C temperature-tested dice

APPLICATIONS

Wireless base station control circuits
Optical network control circuits
Instrumentation
Sensors and controls
- **Thermocouples**
- **Resistor thermal detectors (RTDs)**
- **Strain bridges**
- **Shunt current measurements**

Precision filters

GENERAL DESCRIPTION

The OP07 has very low input offset voltage (75 µV maximum for OP07E) that is obtained by trimming at the wafer stage. These low offset voltages generally eliminate any need for external nulling. The OP07 also features low input bias current (±4 nA for the OP07E) and high open-loop gain (200 V/mV for the OP07E). The low offset and high open-loop gain make the OP07 particularly useful for high gain instrumentation applications.

PIN CONFIGURATION

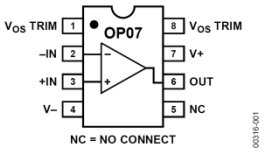
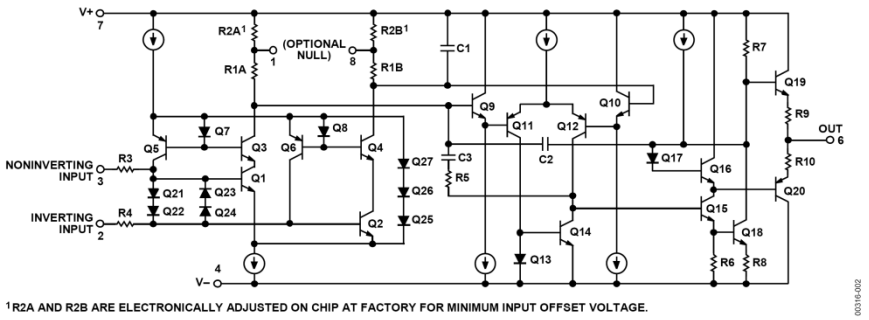


Figure 1.

The wide input voltage range of ±13 V minimum combined with a high CMRR of 106 dB (OP07E) and high input impedance provide high accuracy in the noninverting circuit configuration. Excellent linearity and gain accuracy can be maintained even at high closed-loop gains. Stability of offsets and gain with time or variations in temperature is excellent. The accuracy and stability of the OP07, even at high gain, combined with the freedom from external nulling have made the OP07 an industry standard for instrumentation applications.

The OP07 is available in two standard performance grades. The OP07E is specified for operation over the 0°C to 70°C range, and the OP07C is specified over the −40°C to +85°C temperature range.

The OP07 is available in epoxy 8-lead PDIP and 8-lead narrow SOIC packages. For CERDIP and TO-99 packages and standard microcircuit drawing (SMD) versions, see the OP77.



[1]R2A AND R2B ARE ELECTRONICALLY ADJUSTED ON CHIP AT FACTORY FOR MINIMUM INPUT OFFSET VOLTAGE.

Figure 2. Simplified Schematic

Rev. F

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

National Semiconductor

May 1999

DAC0808
8-Bit D/A Converter

General Description

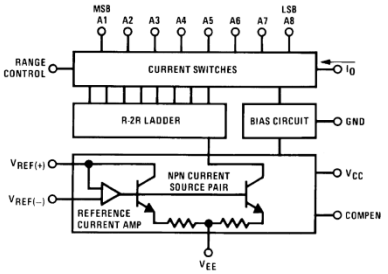
The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 $I_{REF}/256$. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 µA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 is independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

Features

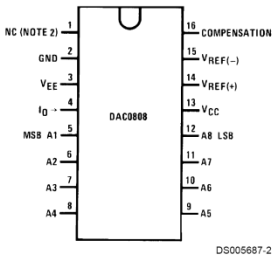
- Relative accuracy: ±0.19% error maximum
- Full scale current match: ±1 LSB typ
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/µs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

Block and Connection Diagrams



DS005687-1

Dual-In-Line Package

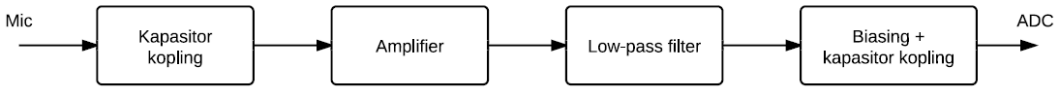


DS005687-2

Top View
Order Number DAC0808
See NS Package M16A or N16A

LAMPIRAN E

PENDEKATAN TRANSFER FUNCTION RANGKAIAN ANALOG



Biasing + kapasitor kopling

Rangkaian Biasing + kapasitor kopling dapat dilihat pada **Gambar 3.7**. Dengan menyederhanakan rangkaian Biasing + kapasitor kopling dihitung, maka didapat *transfer function* sebagai berikut:

$$H_1 = \frac{(R1 + R2 + R3)||R4}{\{(R1 + R2 + R3)||R4\} + \frac{1}{s\,C3}}$$

$$H_1 = \frac{5{,}45\ 10^3}{5{,}45\ 10^3 + \frac{10^9}{47\ s}} = \frac{5{,}45\ 10^3}{5{,}45\ 10^3 + \frac{10^9}{47\ s}}$$

$$H_1 = \frac{s}{s + 3900}$$

Low-pass filter

Rangkaian *low-pass filter* dapat dilihat pada **Gambar 3.5**. *Transfer function low-pass filter* dapat diturunkan dengan menggunakan persamaan (3.1).

$$H_2 = \frac{1}{s^2 2{,}2 \times 10^{-8} + s\ 2\ \times 10^{-4} + 1}$$

Amplifier

Rangkaian amplifier dapat dilihat pada **Gambar 3.3**. Rangkaian amplifier ini hanya memperkuat semua amplituda pada frekuensi yang masuk, sehingga *transfer function* amplifier hanya merupakan penguatan saja.

$$H_3 = A$$

Tahanan input rangkaian amplifier ini adalah 50 KOhm dapat dilihat pada datasheet LM386.

Kapasitor kopling

Nilai kapasitor kopling yang digunakan adalah 47 nF. Dengan mengetahui tahanan input dari amplifier, maka *transfer function* kapasitor kopling dapat ditulis sebagai berikut

$$H_4 = \frac{50\ 10^3}{50\ 10^3 + \frac{10^9}{47\ s}}$$

$$H_4 = \frac{s}{s + 425{,}5}$$

Dengan mengalikan semua *transfer function* maka didapat:

$$H = H_1 \times H_2 \times H_3 \times H_4$$

$$= A \frac{s^2}{(2{,}2 \times 10^{-8} \times s^4) + (1{,}1 \times 10^{-3} \times s^3) + (5{,}36 \times s^2) + (4{,}34 \times 10^4 \times s) + (1{,}6 \times 10^6)}$$

Dengan menggunakan MATLAB, maka respon frekuensi dengan A = 20 dapat dilihat pada gambar berikut.