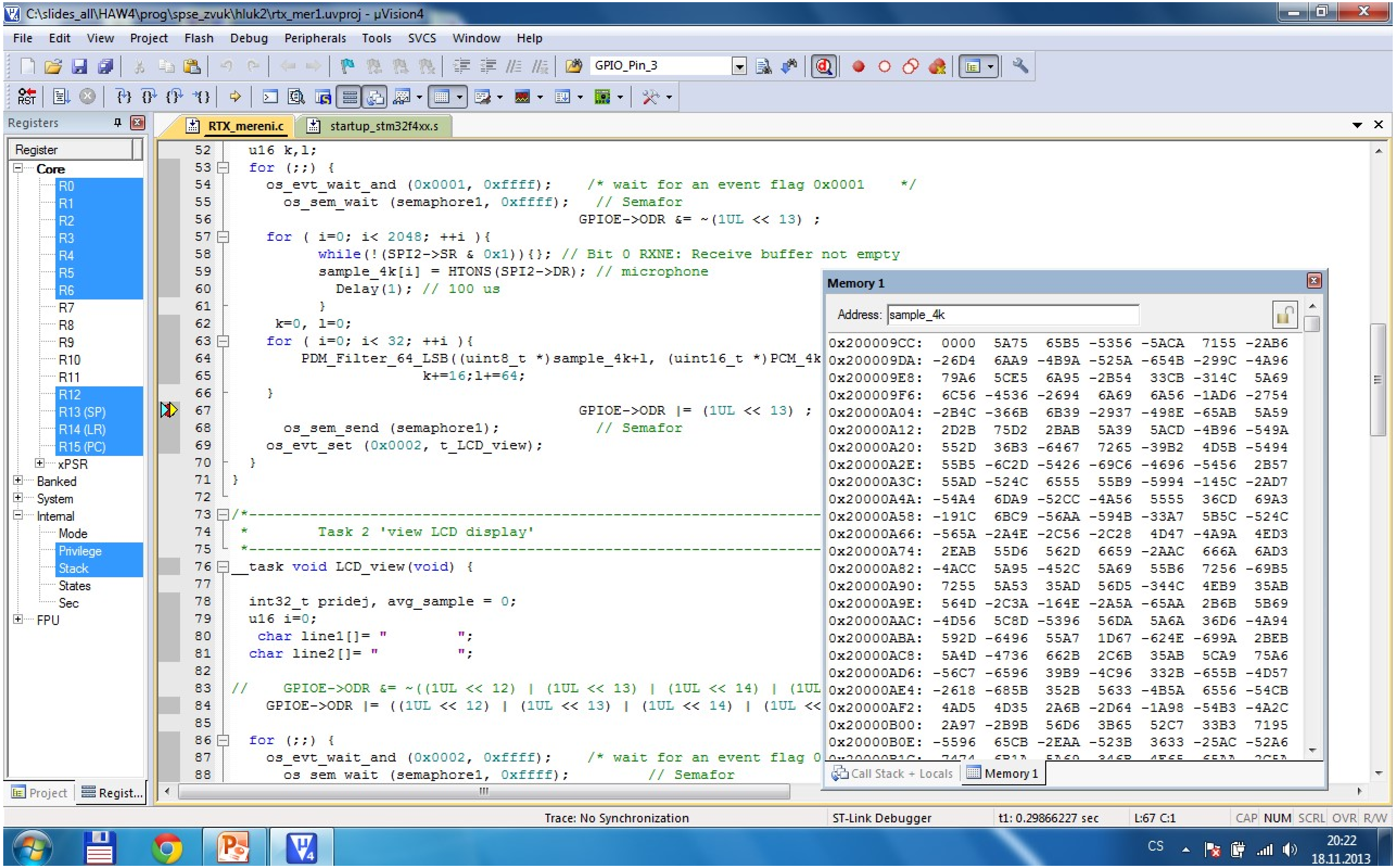
Projekt STM32 Keil v.4 př. 1



Příklad :

Realizujte měřič hluku, intezitu hluku zobrazujte na připojeném LCD displeji.
Rozblikejte LED diody na portu E procesoru STM32 během vzorkování zvuku.
ST MEMS mikrofon (MP45DT02) je digitální, používá PDM protokol .

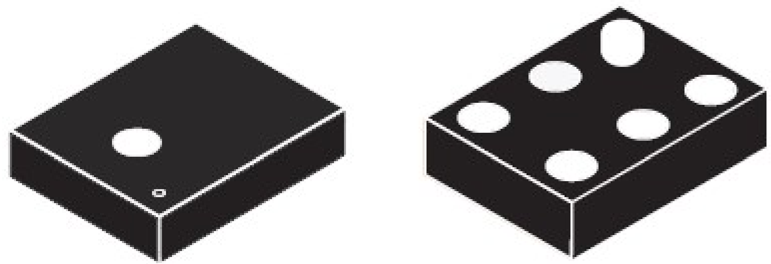
MP45DT02

MEMS audio sensor omnidirectional digital microphone

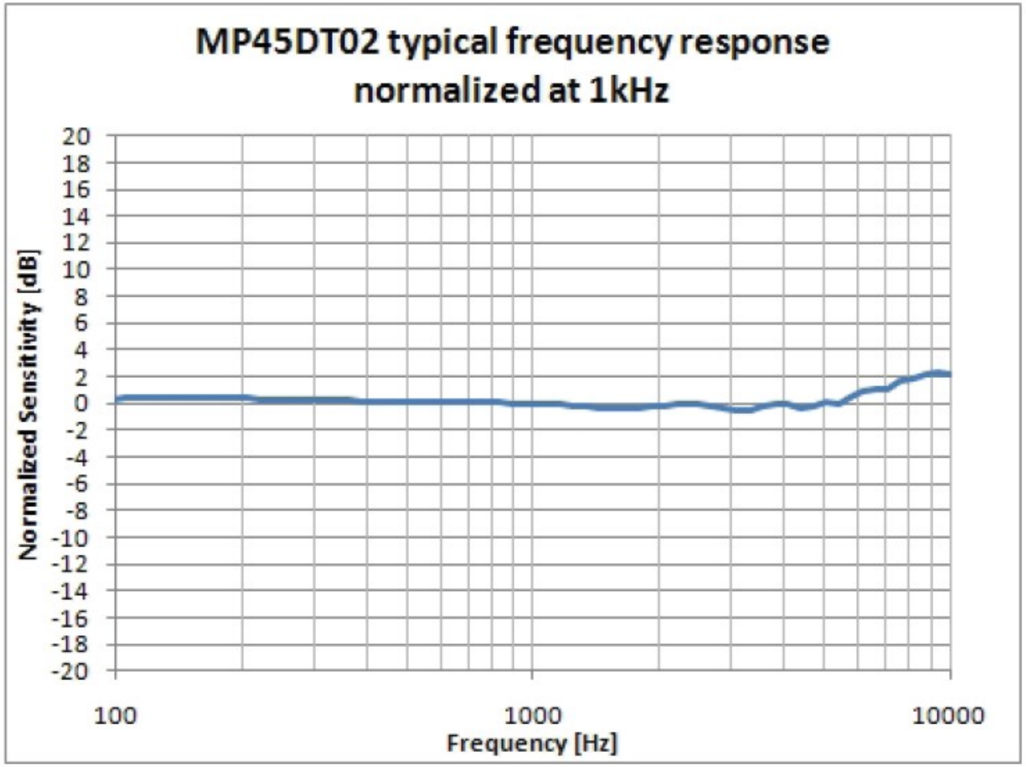
Datasheet — production data

Features

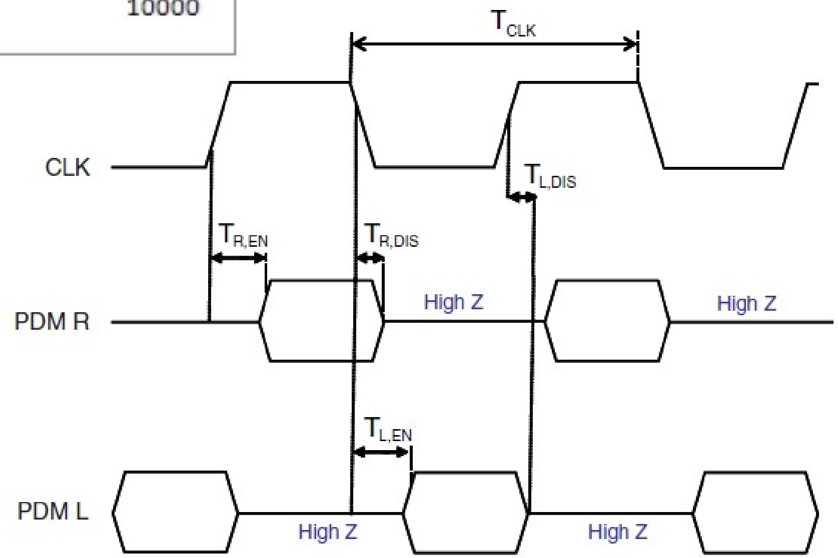
- Single supply voltage
- Low power consumption
- 120 dBSPL acoustic overload point
- Omnidirectional sensitivity
- PDM single-bit output with option for stereo configuration
- HLGA package (SMD-compliant)
- ECOPACK®, RoHS, and “Green” compliant



HLGA 4.72 x 3.76 6LD



MP45DT02 typical frequency response
normalized at 1kHz
20
18
16
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
Normalized Sensitivity [dB]
100
1000
10000
Frequency [Hz]



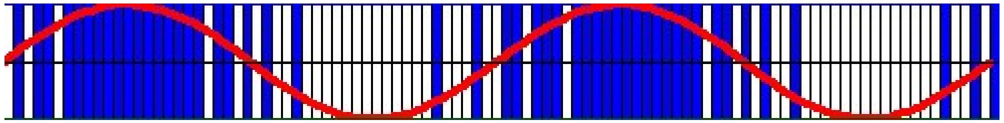
T CLK
CLK
T L,DIS
T R,EN
T R,DIS
PDM R
High Z
High Z
T L,EN
PDM L
High Z
High Z

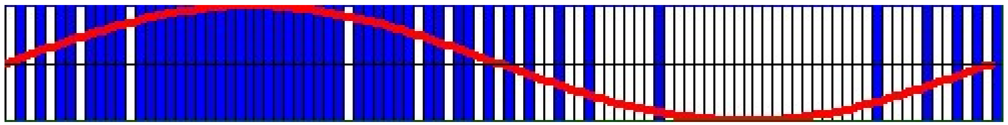
Pulzně-hustotní modulace (Pulse-density modulation)

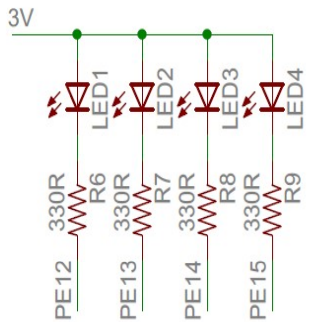
PDM je forma modulace, která prezentuje analogový signál. Relativní hustota pulzů prezentuje amplitudu analogového signálu.

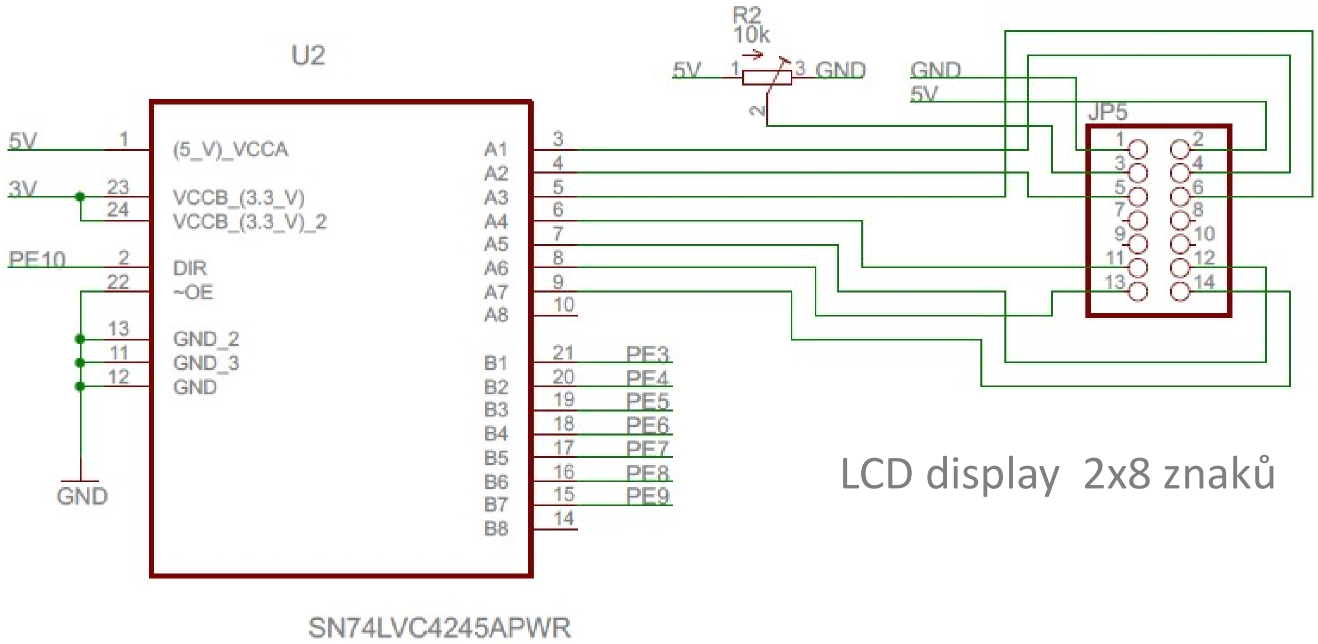
Dvě periody signálu sinus budou vypadat takto :

01011011111111111111011010100100000000000000001000100110111011111111111111101101

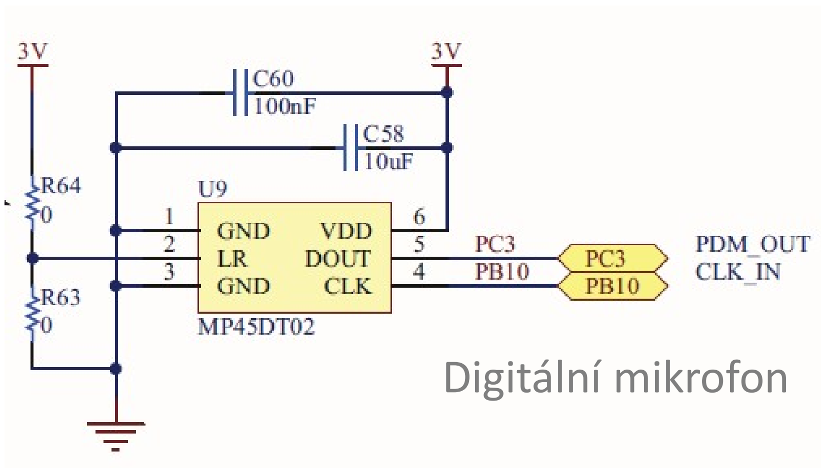




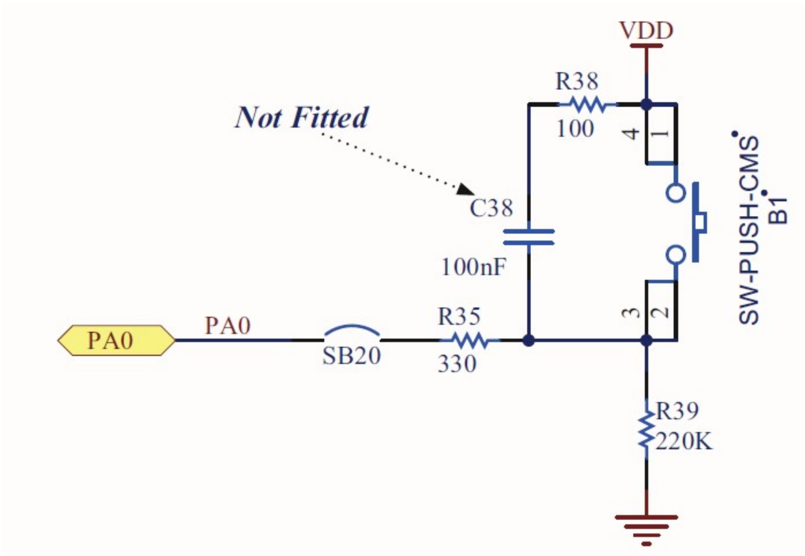




LCD display 2x8 znaků

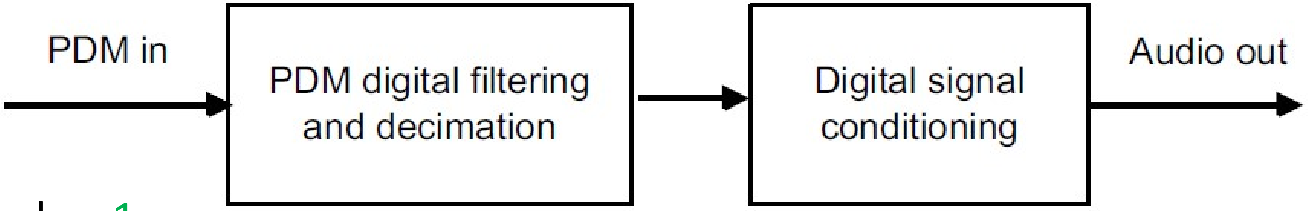


Digitální mikrofon



Aplikace PDM filtru na data z mikrofonu

```
#include "pdm_filter.h“

PDMFilter_InitStruct Filter;

  Filter.LP_HZ = 8000;
  Filter.HP_HZ = 40;
  Filter.Fs = 16000;
  Filter.Out_MicChannels = 1;
  Filter.In_MicChannels = 1;
   PDM_Filter_Init((PDMFilter_InitStruct *)&Filter);
```



The PDM library is composed of a structure and the implementation of four PDM filter functions. The library uses two buffers, the PDM Input buffer and the PCM Output buffer; the application must define these buffers in the main program.

- Input buffer (data) is a uint8 variable with a length equal to (Output frequency / 1000 * decimation factor * Input Microphone Channels / 8) at least.
- Output buffer (dataOut) is a uint16 variable with a length equal to (Output frequency / 1000 * Output Microphone Channels) at least.

The structure is defined in the pdm_filter.h file and is used to configure the filter; it is composed as follows:

Proběhne decimace vstupu z mikrofonu s faktorem 64

decimace	64	
fs	16000	Hz
fmic	1024000	Hz

Výstupem z digitálního filtru je 16-bit hodnota v rozsahu [–32768,32767] pro referenční zesílení 0 dB

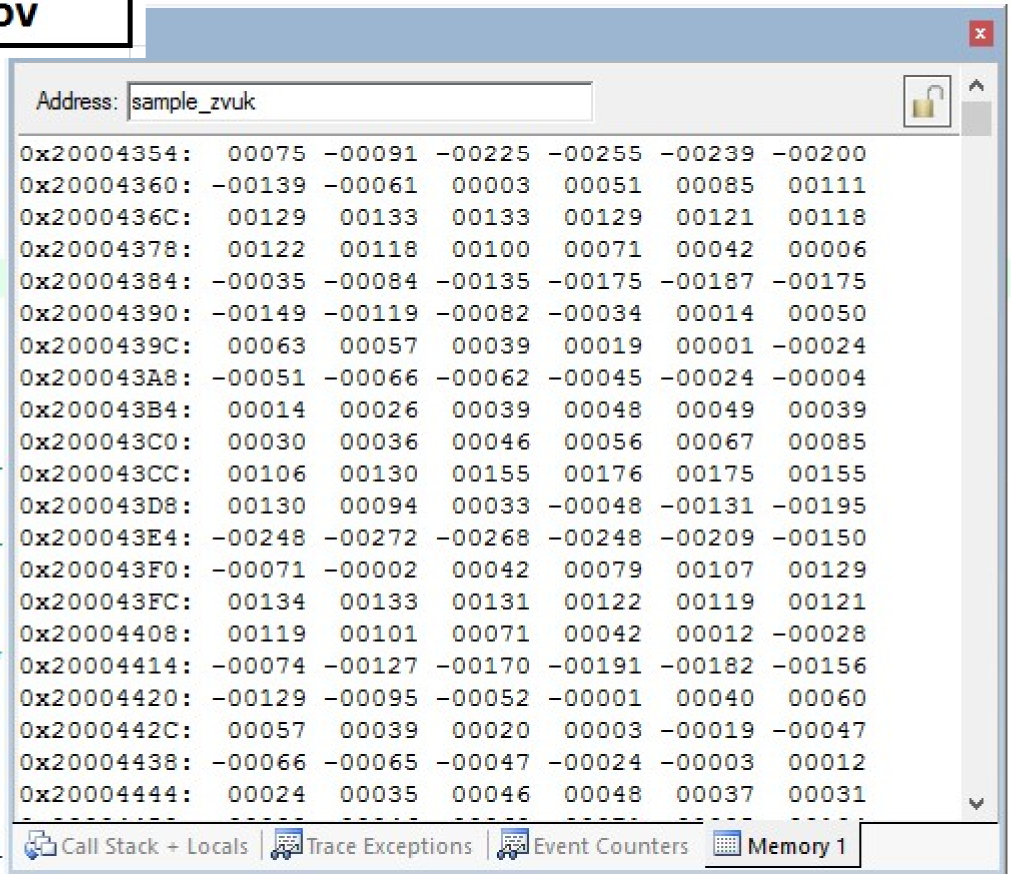
dB	amplituda dig.	hlasitost
0	32768	reference
-24	2048	max PCM
-48	128	tichý zvonek
-76	5	ticho

Úroveň signálu [dB]= 20*log (signal_PCM) /32768)

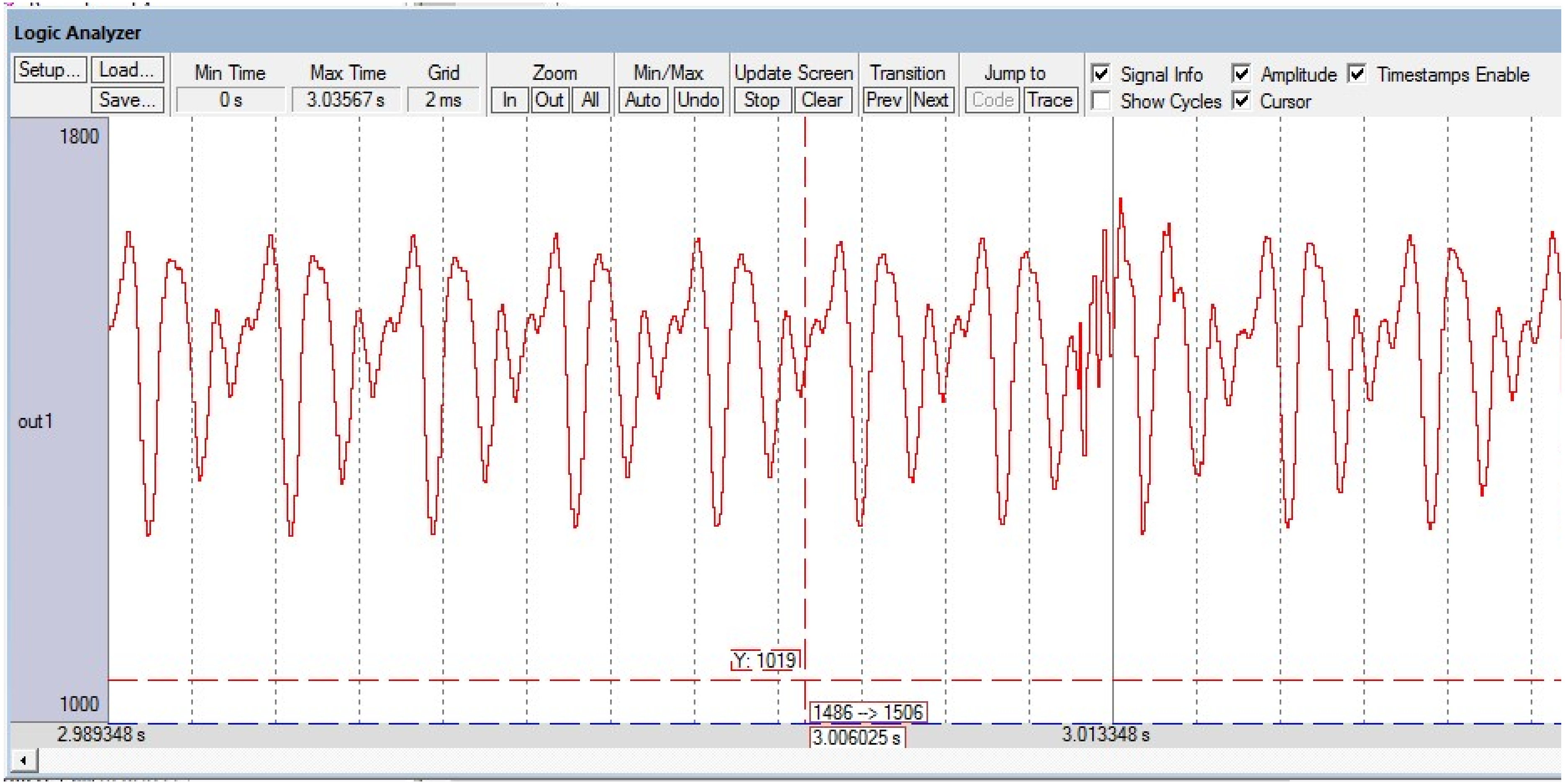
Parametry vstupního i výstupního bufferu výpočteme z uvedeného předpisu :

Fs výstupní frekvence	**16000**	**Hz**
faktor decimace	64	
vstupní kanal	1	
výstupní kanal	1	
délka in bufferu	**128**	**8-bit slov**
délka out bufferu	**16**	**16-bit slov**

Záznam samohlásky „a“ střední hlasitosti
Ve formátu PCM :



Záznam samohlásky „a" střední hlasitosti
průchodem 12-bit D/A převodníkem (offset 1500):



Program čtení dat z mikrofonu dig. úprava

```
const uint16_t    N=8192 ;
uint16_t   sample_mic[N] ;
int16_t     sample_zvuk[N>>3] ;


for ( i=0; i< 8192; ++i )
    {
     while(!(SPI2->SR & 0x1)){};
             sample_mic[i] = HTONS(SPI2->DR);
             }
k=0, l=0;
for ( i=0; i< N>>7 ; ++i ){
PDM_Filter_64_LSB((uint8_t *)sample_mic+l, (uint16_t *)sample_zvuk+k, 1 ,
(PDMFilter_InitStruct *)&Filter);
             l+=128;   k+=16;    }
```

Konfigurace pinů pro napojení LCD displeje

```
void LCD_config(void){
                              RCC->AHB1ENR  |= ((1UL <<  4) );          //  PE povolit clock
GPIOE->MODER    |=  (               (1UL << 2*3) |                      //  RS
                             (1UL << 2*4) |                             //  R/W
                             (1UL << 2*5) |                             //  E
                              (1UL << 2*6) |                            // vystup DB4..DB7 data
                             (1UL << 2*7) |  (1UL << 2*8) | (1UL << 2*9) | (1UL << 2*10))          ;
GPIOE->OTYPER   &= ~( (1UL << 3) |(1UL << 4) |(1UL << 5) | (1UL <<   6) |
                                    (1UL <<   7) | (1UL <<   8) | (1UL <<   9) |(1UL <<  10));   // Push - Pull
GPIOE->OSPEEDR    &= ~( (3UL << 2*3) | (3UL << 2*4) | (3UL << 2*5 )|(3UL << 2*6) |(3UL << 2*7) |
                                     (3UL << 2*8) | (3UL << 2*9) |(3UL << 2*10) ) ;
GPIOE->OSPEEDR  |=  (               (2UL << 2*3) |(2UL << 2*4) | (2UL << 2*5) |         // 50 Mhz out
                             (2UL << 2*6) | (2UL << 2*7) |  (2UL << 2*8) | (2UL << 2*9) |(2UL << 2*10) );
GPIOE->PUPDR    &= ~((3UL << 2*3) |(3UL << 2*4) |(3UL << 2*5) |(3UL << 2*6) |
                                     (3UL << 2*7) | (3UL << 2*8) | (3UL << 2*9) |(3UL << 2*10) )  ;}
```

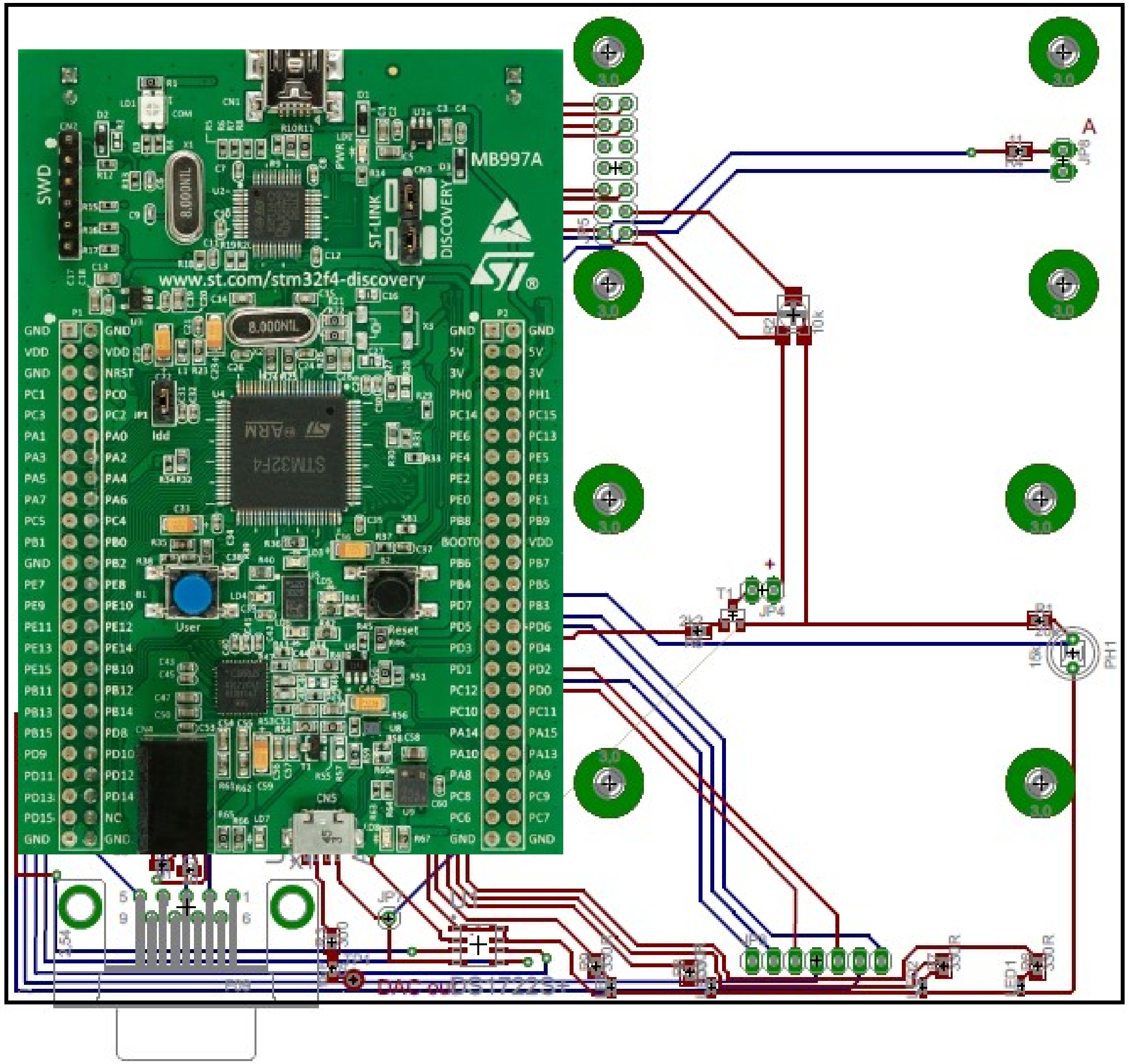
Podprogram zápis 4-bit do reg. LCD

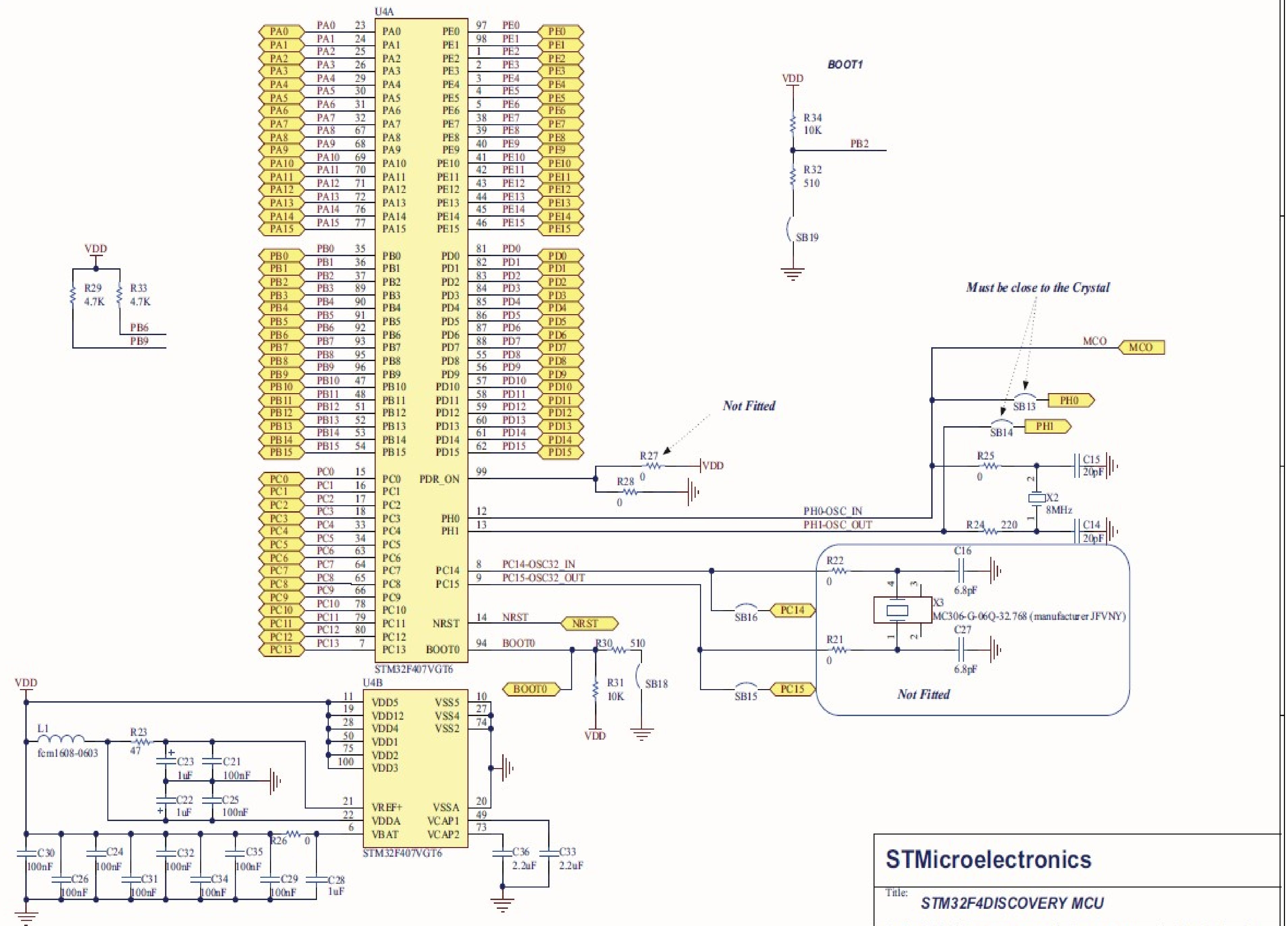
```
void write_nibble_res(uint16_t nibble){
        uint16_t x;
        GPIOE->BSRRH  = (   (1UL <<  (3)) |        //          RS=0
                (1UL << ( 4) ) |        //          R/W=0
                (1UL << (5)) );         //          E=0
        Delay(1);  // 100 us
        GPIOE->BSRRL = (1UL << 5) ;    //          E=1  (clk)
        nibble &= 0x0F;
        nibble = nibble << 6 ;
        x = GPIOE->ODR & 0xFC3F ;                  // nuluj data
        GPIOE->ODR = nibble | x ;                  // pridej data

        Delay(1);  // 100 us
        GPIOE->BSRRH = (1UL << (5)) ;              //          E=0
        Delay(1);  // 100 us

}
```

Program zobrazování na LCD displeji

```
 sprintf (line1, "Hluk dB ");
puts_LCD (1,line1);
GPIOE->ODR |= (1UL << 12) ;
avg_sample = 0 ;
for ( i=1; i< 512; ++i ){
		pridej=PCM_4k[i];
		avg_sample = (avg_sample +abs( pridej)) ;}
 avg_sample /= 511;
sprintf (line2,"%8d ",avg_sample-110);
puts_LCD (2,line2);
```

Studentský kit s periferiemi propojenými na STM32F4DISCOVERY



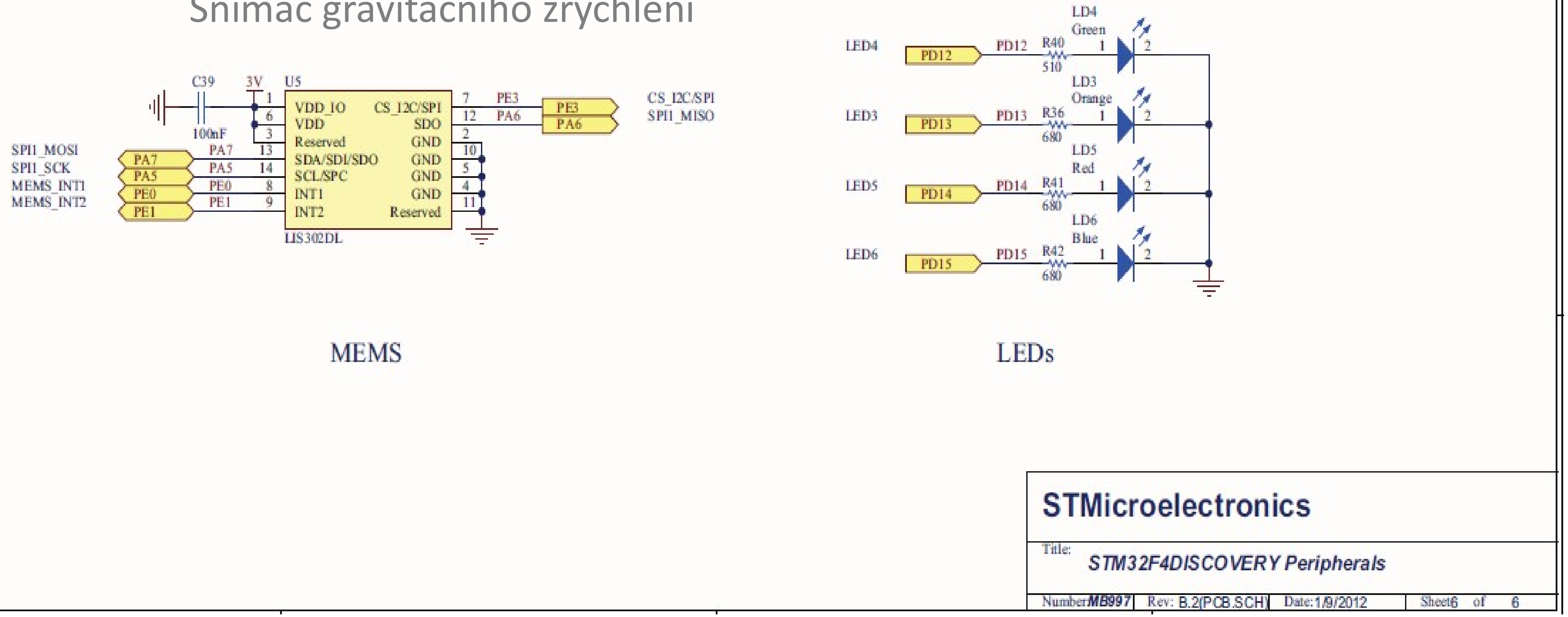


STMicroelectronics

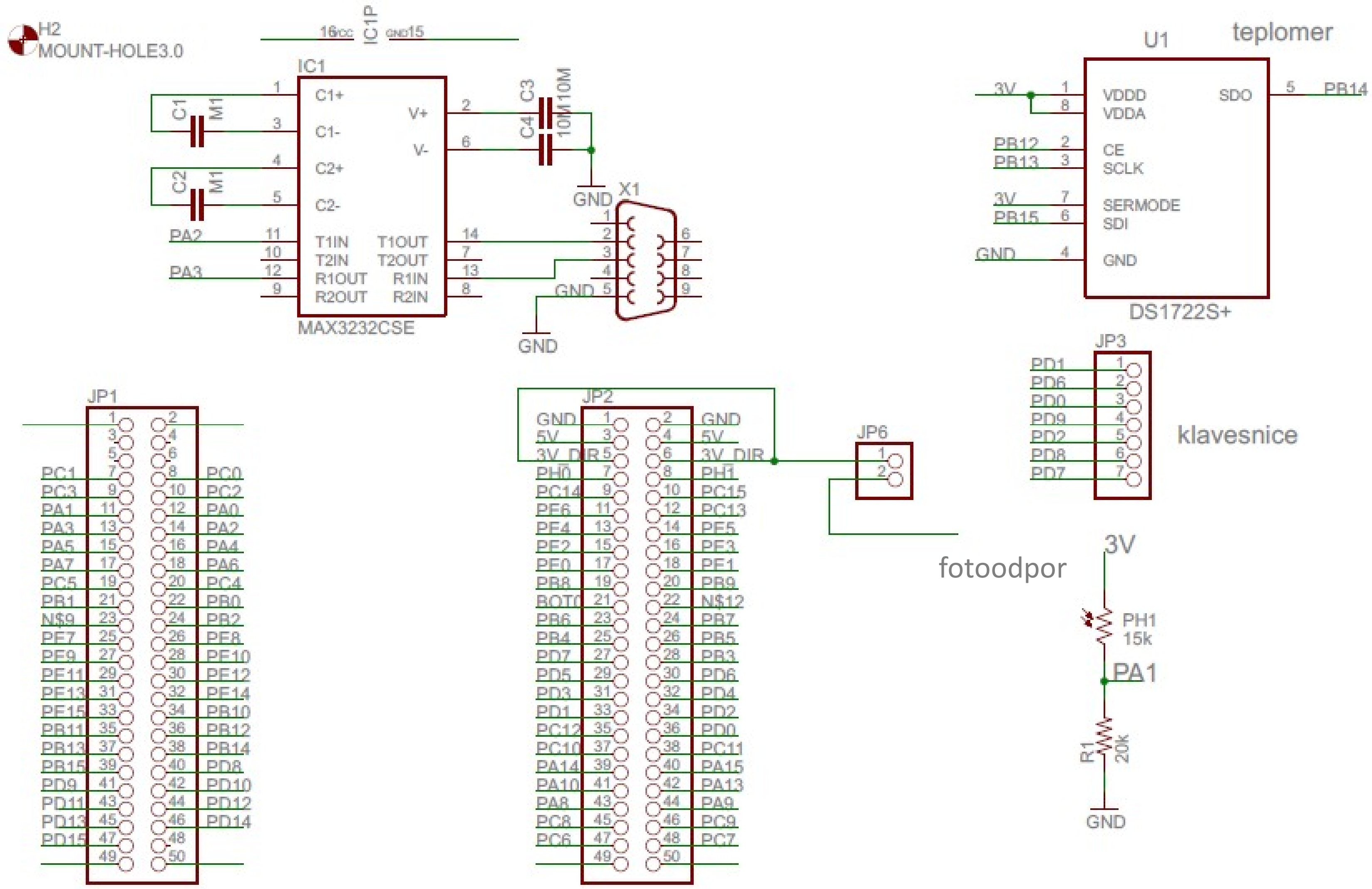
Title: STM32F4DISCOVERY MCU

Number: MB997 | Rev: B.2(PCB.SCH) | Date: 1/9/2012 | Sheet 3 of 6

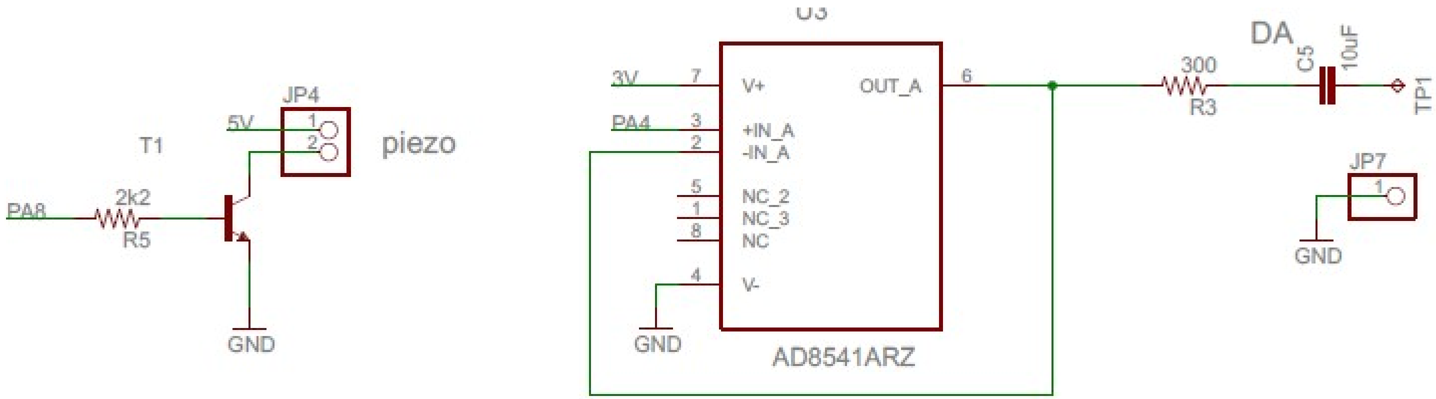
Snímač gravitačního zrychlení



Konvertor RS232



Konektory napojení STM32F4 Discovery



T1
JP4
5V
1
2
piezo
PA8
2k2
R5
GND
U3
3V
7
V+
OUT_A
6
PA4
3
+IN_A
2
-IN_A
5
NC_2
1
NC_3
8
NC
4
V-
GND
AD8541ARZ
DA
300
R3
C5
10uF
TP1
JP7
1
GND

Literatura :

RM0090

Reference manual STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based 32-bit MCUs [online] **www.st.com**

© 2011 STMicroelectronicsstr 136 – 154 [cit. 2013-10-31].