

Detekcia prekážky ultrazvukom

Zapojenie generuje ultrazvukové vlny o frekvencií 40kHz, ktoré po následnom odrazení od prekážky detekuje. Na generovanie signálu slúži mikročip ATMEGA16. Následne signál putuje do integrovaného obvodu MAX232, ktorý signál +5 zmení na -8,5 a 0 na +8,5, tento zosilnený signál je vysielaný ultrazvukovým vysilačom UST40T. Odrazený signál je zachytený na ultrazvukovom prijímači UST40R, tento signál však treba zosilniť operačným zosilňovačom LM1458, aby sme ho boli schopní následne detekovať komparátorom LM311, ktorý rozsvieti led-diódu.

Zapojenie a využitie mikroprocesora ATMEGA16:

Využiť mikroprocesor len na účel generovania ultrazvukovej frekvencie je zbytočne komplikované, ale zvolil som túto možnosť pre neskoršie možné vylepšenie a prerobenie zapojenia (meranie vzdialenosti prekážky, nahradenie led-diód displejom, s možnosťou využitia ako parkovací senzor). Samotný mikroprocesor treba napájať na +5V (10) a zem (11) napojiť kristal (12,13) generujúci mikroprocesoru časové kroky. Najzložitejšie ho bolo správne naprogramovať, a to cez paralelný port počítača pomocou programátora stk200, inak je jeho zapojenie pomerne jednoduché.

Integrovaní obvod MAX232:

Signál z mikroprocesora je invertovaný a zosilnený týmto obvodom. Okrem toho tento obvod vyrába napájanie (-8,5 a +8,5 V) pre operačný zosilňovač LM1458 a komparátor LM311. Zapojil som ho podľa odporúčaného zapojenia (viď referencie) s použitím kondenzátorov s kapacitou jeden mikrofaraď.

Ultrazvukový vysilač a prijímač UST40R/T

Vysilač (UST40T) bol jednoducho pripojený na výstup MAX232 (v. 14 a 7) a prijímač je jedným koncom uzemnený a druhým pokračuje cez kondenzátor a odpor na operačný zosilňovač LM1458.

Zosilňovač LM1458:

Vzhľadom na to že väčšina energie z vysilača sa rozptýli do okolia a len jej nepatrný zlomok sa zachytí na prijímači je treba prijatý signál dostatočne zosilniť a to správnym na volením pracovných odporov.

Komparátor LM311:

Výstup zo zosilňovača išiel cez polovodičovú diódu a nabíjal kondenzátor, čo bolo pripojené na komparátor a porovnávané z referenčným napätím +0,1V, ktoré som získal z +5v a použitím odporovej deličky napätia v pomere 1/50, ak napätie bolo väčšie ako referenčné dióda sa rozsvietila.

Vylepšenia zapojenia:

Lepšia citlivosť sa dá získať lepším zosilnením (použitím viacerých zosilňovačov, alebo ich lepšie nastaviť), alebo znížením referenčného napätia komparátora (delička vo väčšom pomere). No aj tieto opatrenia majú svoje limity, ak sa stane zariadenie príliš citlivé môže byť rušené okolitým šumom, môže detekovať prekážky aj vo väčšej vzdialenosti ako je žiaduce, alebo aj vybráciami samotného zariadenia. Lepší dosah sa dá tiež dosiahnuť zvýšením výkonu vysilača, ale musíme si dať pozor, aby sme ho nepreťažili. Použitím mikročipu sa nám otvára viacero možností ako tento detektor využiť: Mikročip môže merať čas odrazu signálu a následne určiť vzdialenosť prekážky, čo môžeme použiť napr. pri výrobe parkovacích senzorov. Taktiež môže merať čas, počas ktorého je prekážka v jeho dosahu, alebo počítať počet prekážok, ktoré prešli pred senzorom. Nasledovné údaje sa dajú previesť na displej. Na to sú však potrebné ďalšie úpravy zapojenia a naprogramovania mikroprocesoru.

Vypracoval :Jakub Maxa

referencie:

ATMEGA16 <http://www.atmel.com/Images/doc2466.pdf>

MAX232 <http://www.ti.com/lit/ds/symlink/max232.pdf>

LM1458 <http://www.ti.com/lit/ds/symlink/lm1458.pdf>

LM 311 www.fairchildsemi.com/ds/LM/LM311.pdf

programovanie mikročipov http://svetelektro.com/Download/avr_kniha.pdf

námet na celú prácu: <http://www.robot-electronics.co.uk/html/srf04tech.htm>

prílohy

```
program mikroprocesora
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRD = 0x13; //nastavenie pinov ako vystupne (kontrolna led, vysielanie 40kHz)
    PORTD = 0x10; //zapnutie kontrolnej led

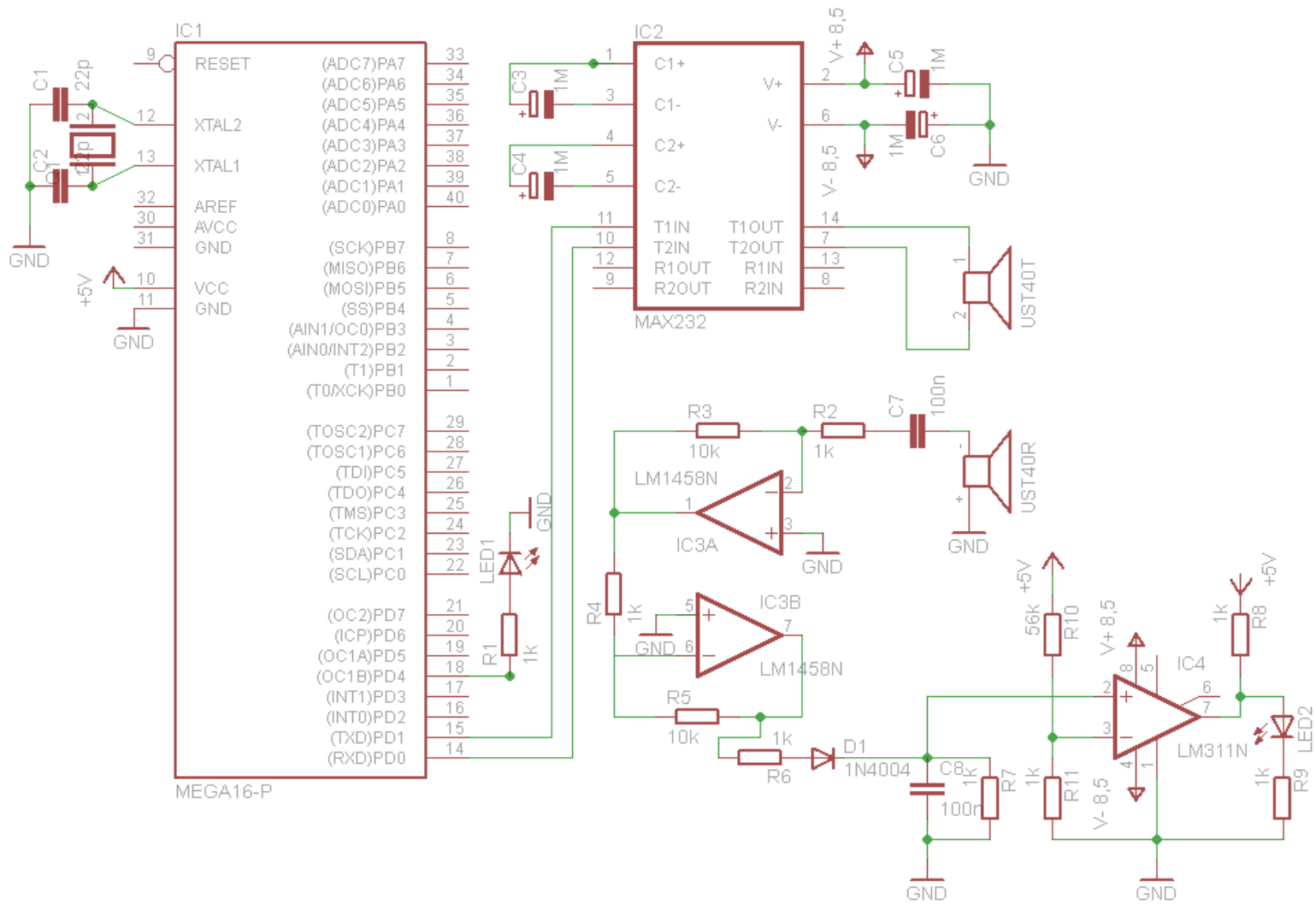
    long cyklk = 0; //citac cyklov

    int ledOn = 0x10; //rozsvietenie kontrolnej led
    int vysOn = 1; //zapnutie pinu 0 na porte D
    int vysOff = 2; //zapnutie pinu 1 na porte D

    while(1)
    {
        cyklk++;
        if(cyklk==10000)
        {
            ledOn = 0; //vypnutie kontrolnej led
        }
        else
        if(cyklk==40000)
        {
            ledOn = 0x10; //rozsvietenie kontrolnej led
            cyklk = 0;
        }

        _delay_us(12);
        PORTD = vysOn | ledOn; //nastavenie vystupnych pinov
        _delay_us(13);
        PORTD = vysOff | ledOn; //nastavenie vystupnych pinov
    }
    return 0;
}
```

schéma zapojenia



ako to vyzerá

