

Mikro počítačové Systémy MIPS

(Microcontrollers)

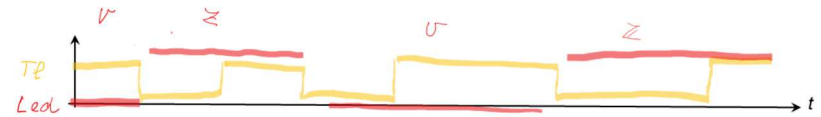
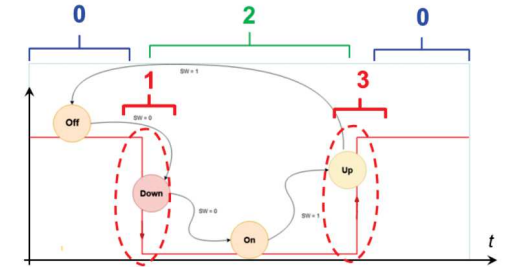
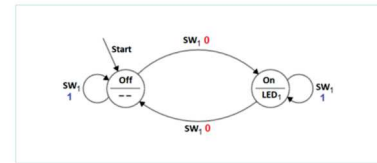
Prednáška 4 / 2023 Binárne vstupy.

Treba niečo „počítať“, ak výstupom kontaktu je len Zap/Vyp?
Derivácia (diferencia) je tu použitá implicitne – explicitne.

1

Analýza cv.03.

Cieľ: Naprogramovať zapínanie a vypínanie led jedným tlačítkom.



2

Stav ledky sa zmení pri zatlačení (1), niekde v strede (2) alebo pri pustení tlačítka (3)? Ak sa zhodneme, že zmena stavu sa udeje v (1), resp. (3) potom je namieste kde a akým spôsobom treba pozmeniť vzorový program aby sa prepínanie stavu led zmenilo: (1)->(3), resp. (3)->(1).

Odozvané programy použili funkciu `_delay_ms(10)`:

- Nula krát,
- Dva krát, resp.
- Štyri krát.

Namieste je otázka: koľko trvá stav (1), resp. (3) pri zatlačení, resp. pustení tlačítka?

Odpoveď:

1. Niečo menej ako 2us.
2. O niečo viac ako 10ms.

Skúste zdôvodniť.

Na prednáške sme na ošetrenie zákmitov použili „diferenciu“. Môžeme povedať, že príklad z prednášky je stavový automat a naopak príklad z cvičenia vyhodnocuje diferenciu, teda smer? Down, resp. Up? Takto to bolo v roku 2022 ZS.

LEDKA: `if ("TRUE") toggle_bit(PORTB,LED1);`

```

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#define set_bit(ADDRESS,BIT) (ADDRESS |= (1<<BIT))
#define toggle_bit(ADDRESS,BIT) (ADDRESS ^= (1<<BIT))
#define LED1 PB5 // zabudovana dioda
#define SW1 PD6 // tlacitko
enum states { Low, High };
int main(void){
enum states last_st = High;
enum states new_st = High;
set_bit(DDRB,LED1); // set pin LED1 as output
set_bit(PORTD,SW1); // pull-up resistor ON
while(1){
delay_ms(10); // chvilu počkajme
{ last_st = new_st;
if (bit_is_clear(PIND,SW1)) new_st = Low; else new_st = High;
// if (new_st && !last_st) toggle_bit(PORTB,LED1);
if (!new_st && last_st) toggle_bit(PORTB,LED1);
} /* end of while loop */
return(0); // sem nikdy neprideme
}

```

TLAČÍTKO: $\Delta = y(t_{n+1}) - y(t_n)$

3

Zadefinujeme dva stavy Ledky:

Z – zapnutá

V – vypnutá.

A prechod medzi týmito stavmi realizujeme NÁBEŽNOU, resp. DOBEŽNOU hranou.

I/O Príklad: jednoduchý kontakt

Netreba pripájať žiadne externé pullup rezistory

Je to pravda? Platí to vždy a všade?

4

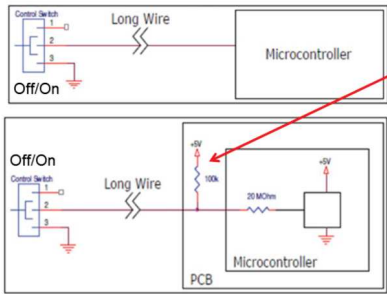
Ešte na prvej prednáške sme si ukázali, aké je to jednoduché, keď chceme realizovať binárny vstup. Nakonfigurujeme daný pin portu ako vstupný, a je to. Prax niekedy od nás požaduje tzv. vzdialený zdroj signálu. Za takéto môžeme považovať aj pripojenie kontaktu viac ako 5 m dlhým vedením.

Tento príklad predpokladá napájanie procesora VCC = 5V. Vstupnú impedanciu pinu niekoľko MOhm. Tento údaj nenájde v KL, ale nájdeme tam Input Leakage Current I/O Pin (Low/High) max. value 1uA.

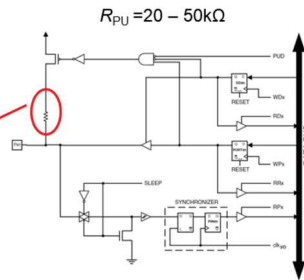
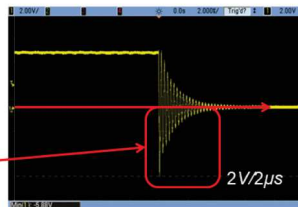
Keďže jeden krát zažiť je viac ako niekoľko krát čítať pokúsím sa zopakovať príklad z internetu. Musel som ho trochu modifikovať. Dôvod: Doma nemám až tak kvalitné vybavenie. Našiel som len 5m tzv. telefónnej dvojlinky a osciloskop s parametrami: 10 MHz a vstupnou impedanciou sondy 1MOhm.

Ochrana I/O vstupov mikropočítača

Predpokladajme pripojenie I/O pinu k tlačítku niekoľko metrov dlhým vodičom.



Takýto priebeh by mohol zničiť mikropočítač

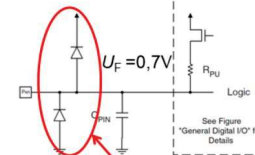
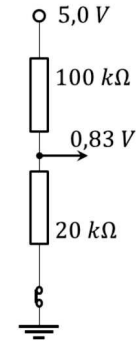
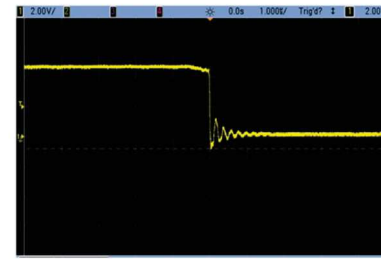
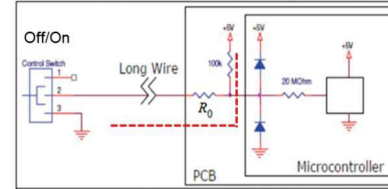


5

AVR majú zabudovaný pullup 20-50 kOhm. Pri niektorých typoch MMP je to aj 100 kOhm. Cez takýto rezistor napájame linku, ktorá sa javí ako RLC záťaž. V prípade aktivovania tlačítka vznikne prechodný proces, ktorý môže zničiť vstupné obvody MMP.

5

Ochrana I/O vstupov mikropočítača: Obmedzenie prúdu.



V ideálnom prípade takáto ochrana postačuje. Avšak ak je napätie veľké alebo ak trvá dlho, môžu sa diódy zničiť, dokonca aj vnútorné obvody mikroprocesora ... Dokonca aj keď sa diódy nezničia, veľké ESD impulzy môžu vyvolať **veľké prúdy v napájaní** mikroprocesora. Toto je ďalší možný zdroj porúch. To znamená, treba obmedziť prúd. Pridáme do vstupu rezistor R_0 . R_0 je treba navrhnuť tak, aby na ňom nevznikol veľký úbytok napätia. Rezistory 100kΩ a R_0 tvoria napäťový delič.

6

Rezistory 100 kOhm a 20 kOhm tvoria odporový delič.

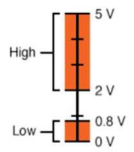
Ak by sme napr. zvolili $R_0 = 20$ kOhm ($2 \cdot 10$ kOhm), zopnutému kontaktu by odpovedalo napätie 0,83V, čo je oveľa viac ako TTL log. úroveň LOW.

Ako je vidieť z obr. dole vľavo, napätie na vstupe neklesne pod hodnotu cca – 0,7V.

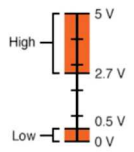
6

TTL

Acceptable TTL Gate Input Signal Levels

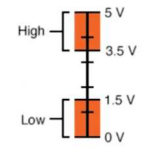


Acceptable TTL Gate Output Signal Levels

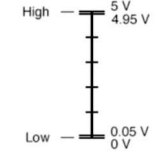


CMOS

Acceptable CMOS Gate Input Signal Levels

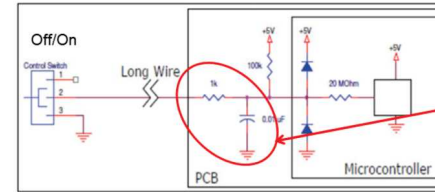


Acceptable CMOS Gate Output Signal Levels



7

Ochrana I/O vstupov mikropočítača Filtrácia



Kvalitnejší spôsob ochrany získame, ak pridáme kondenzátor.

Vytvoríme RC filter.

Obvod obmedzujúci prúd sme upravili na dolnopriepustný filter.

Pri výbere rezistora a kondenzátora musíme zobrať do úvahy frekvenčné vlastnosti obvodu. Filter nesmie odfiltrovať užitočné signály.



FILTER ⇒ pásmo priepustnosti

8

Ináč povedané, mali by sme uvažovať prenosové vlastnosti vstupného kanála. Len málokedy sa v KL objavuje pojem „pásmo priepustnosti“. Namiesto neho prax používa pojmy Rise Time, resp. Fall Time. T.j. ako rýchle signál prejde z jednej úrovne do druhej. A samozrejme nami navrhnuté zariadenie by malo vedieť tieto signály sledovať. Nemalo by ich odfiltrovať.

BandWidth – Rise/Fall Time

LM358
LOW POWER DUAL OPERATIONAL AMPLIFIERS

Features

- Wide bandwidth (unity gain): 1MHz (temperature compensated)

DRV5055 Ratiometric Linear Hall Effect Sensor

1 Features

- Fast 20-kHz Sensing Bandwidth



AD 10 bits \equiv 5mm

Volný pád: $h(t) = h_0 - \frac{1}{2}gt^2 \Rightarrow$

$$t_{1 \text{ bit}} = \sqrt{\frac{0.005}{2^{10}} \frac{2}{g}} \approx 0.99 \text{ ms}$$

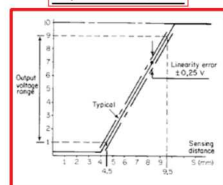
$$t_{10 \text{ bit}} = \sqrt{\frac{0.005}{2^0} \frac{2}{g}} \approx 32 \text{ ms}$$

Proximity Sensors

30 mm (1.18 in.)



Response Time 1 V/msec

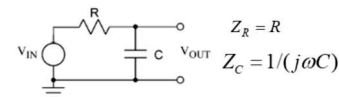


9

Niektoré aplikácie majú uvedené pásmo prepuštnosti, iné čas nábehu – prechodu..

Ochrana I/O vstupov mikropočítača

Súvis medzi „BandWidth – RiseTime“ = „Šírka pásma – Doba nábehu“



Prenos napätového deliča je:

$$H(j\omega) = \frac{V_{OUT}}{V_{IN}} = \frac{Z_C}{Z_C + Z_R} = \frac{1}{1 + j\omega RC}$$

Dosaďme: $\omega = 2\pi f$ a $T = RC$

$$\text{Potom: } H(f) = \frac{1}{1 + j2\pi f RC}$$

Frekvencia f na ktorej modul $H(f)$, t.j.

$$|H(f)| = 1/\sqrt{2} \text{ poklesne o } -3\text{dB}$$

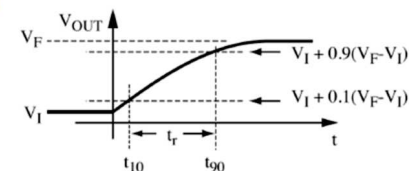
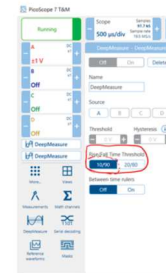
sa nazýva pásmo priepustnosti.

„RiseTime“ je čas za ktorý prejde V_{OUT} z $0.1(V_F - V_I)$ na $0.9(V_F - V_I)$.

$$\text{Výsledkom je: } t_r = 2.2T$$

$$\text{Po dosadení: } T = RC = \frac{1}{2\pi f}$$

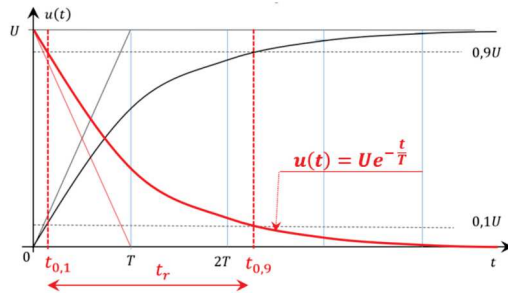
$$t_r = 0.35 \frac{1}{f_{3\text{dB}}}$$



10

Predpokladáme ideálny napätový zdroj s nulovým vnútorným odpor. Napätie na kondenzátore bude nabiehať po exponenciále.

RiseTime“ = „Šírka pásma – Doba nábehu“: Riešenie



„RiseTime“ je čas za ktorý prejde
 V_{OUT} z $0.1(V_f - V_i)$ na $0.9(V_f - V_i)$.

$$0,9U = Ue^{-\frac{t_{0,1}}{T}} \quad \frac{0,9}{0,1} = \frac{e^{-\frac{t_{0,1}}{T}}}{e^{-\frac{t_{0,9}}{T}}} \quad \ln 9 = (t_{0,9} - t_{0,1})\frac{1}{T} = t_r \frac{1}{T}$$

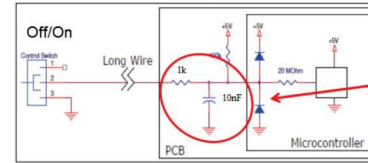
$$t_r \approx 2,2T \quad T = RC \quad \omega = 2\pi f$$

$$t_r = 0,35 \frac{1}{f_{3dB}}$$

11

Zhrnutie: Ak máme napr. uvedené, že tzv. Rise Time zdroja signálu je 10 uS, potom by sme mali zvoliť pásmo priepustnosti napr. 5*väčšie.
 T.j. $f_{3dB} = 0,35/(2us) = 175 \text{ kHz}$. A potom $R*C = 1/(2*\pi*175000) = 0,91 \text{ us}$.
 Ak zvolíme $R = 1 \text{ kOhm}$, potom C (približne) = 1 nF.

Ochrana I/O vstupov mikropočítača
 Filtrácia, pokračovanie



Pridanou hodnotou

RC filtra

je skutočnosť, že RC filter potlačí náhodné, neželané krátke vstupy, ktoré by mohli spôsobiť chybnú prácu mikropočítača.

Poznamenajme, že pre veľké ESD a dlhé vedenia ani toto nemusí postačovať.

Ochrániť vnútorné diódy mikropočítača môžeme pomocou externých **Schottky-ho diód**.

Predpokladajme $t_r = 20 \mu s$

To odpovedá $f_{3dB} = 0,35 \frac{1}{20 \mu s} = 17,5 \text{ kHz}$

$t_r \approx 2,2T = 2,2RC$

Zvolíme $R = 1 \text{ k}\Omega$

potom $C = \frac{t_r}{2,2R} = \frac{20 \mu s}{2,2 \cdot 1 \text{ k}\Omega} = 9,1 \text{ nF}$

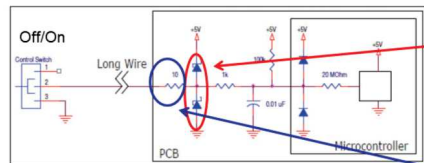
a zaokrúhlime $C \approx 10 \text{ nF}$

12

Príklad: Je dané t_r a vypočítajme R a C .

Ochrana I/O vstupov mikropočítača

Filtrácia, Schottky-ho diódy



Schottky-ho diódy

sa používajú preto, že vedú prúd skorej ako vnútorné ochranné diódy. (úbytok Schottky-ho diód v priepustnom smere je cca 0,2 V. Úbytok vnútorných diód je cca 0,7 V).

Malý sériový rezistor

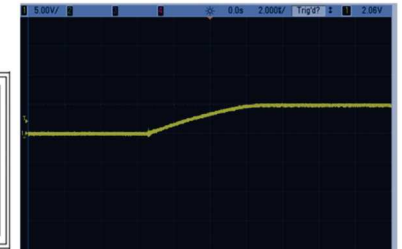
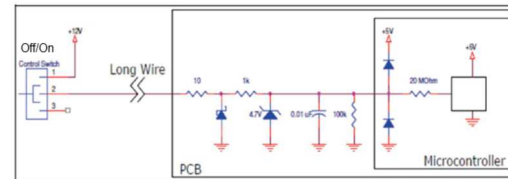
je použitý na ochranu Schottky-ho diód pred veľkým prúdom. Ak sú diódy v stave zopnutom len krátko, postačuje rezistor veľkosti 10 Ω .



13

Ochrana I/O vstupov mikropočítača

Iné možné riešenia



I/O pin je cez tlačítko pripojený na napätie 12V.

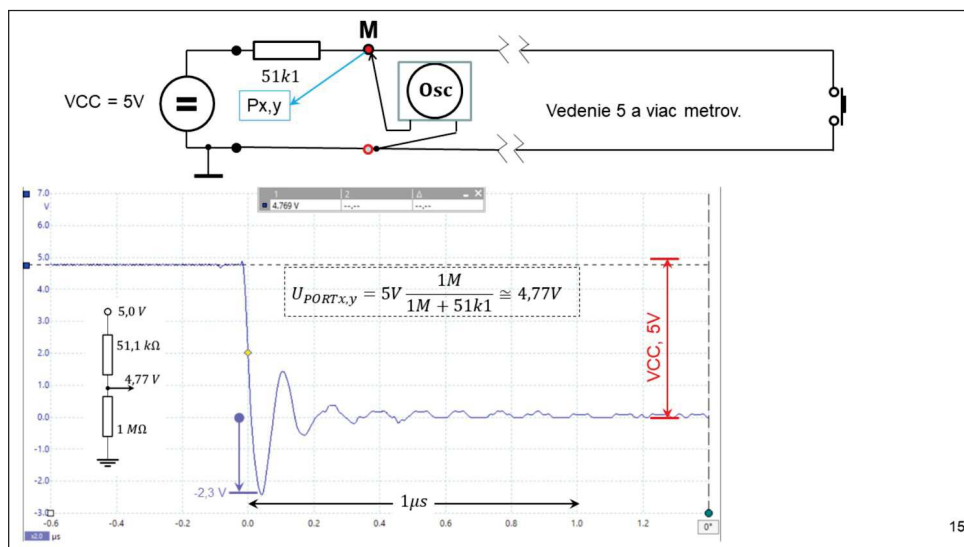
V tomto prípade kontakt nespína I/O pin k zemi, ale k 12V-om.

Schottky-ho dióda obmedzuje záporné napätové špičky.

Zenerová dióda + rezistor (obmedzuje prúd) = zdroj napätia „log. 1“ – cca 5V.

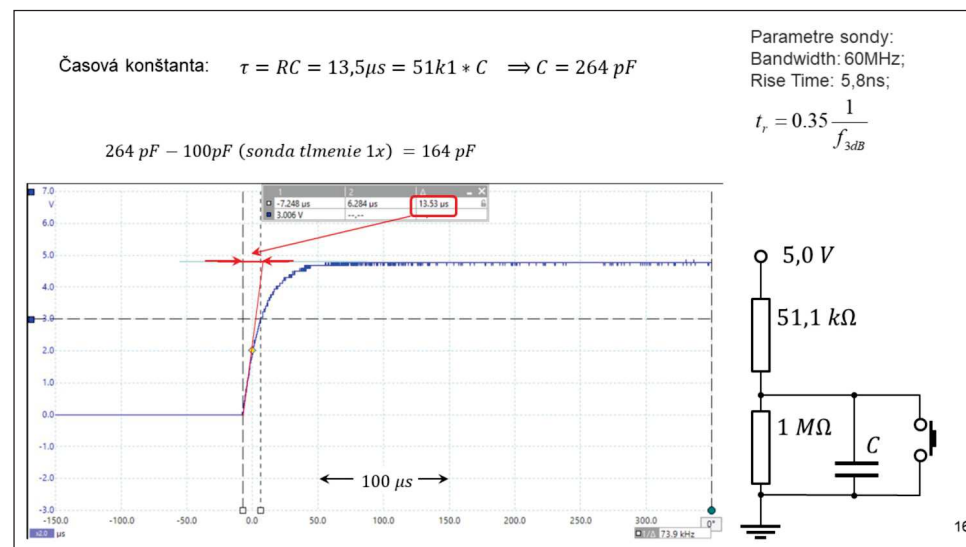
Poznamenajme, že rezistor obmedzujúci prúd treba voliť dostatočne malý (cca 1 mA) na jednej strane, a na strane druhej zas tak veľký, aby sa zenerová dióda dostala do vodivej oblasti.

14



Príklad sme zopakovali, a trochu modifikovali. Použili sme tvrdý zdroj 5V. Namiesto tlačítka sme len vzdialené konce dvojlínky skratovali. Do plus napájania sme pripojili rezistor 51k1.

Vnútny odpor sondy osciloskopu a 51k1 rezistor vytvorili odporový delič. V bode M sme odmerali pokles napätia na hodnotu 4,77V. T.j. zhoda teórie a praxe. Pri skrate vznikol prechodný proces odpovedajúci sústave 2. rádu kmitavej tlmenej.

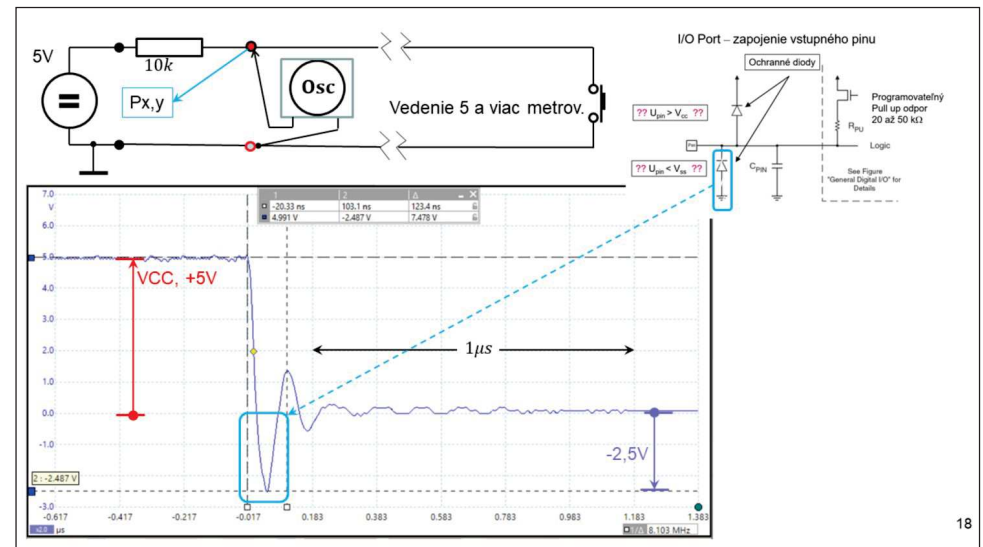
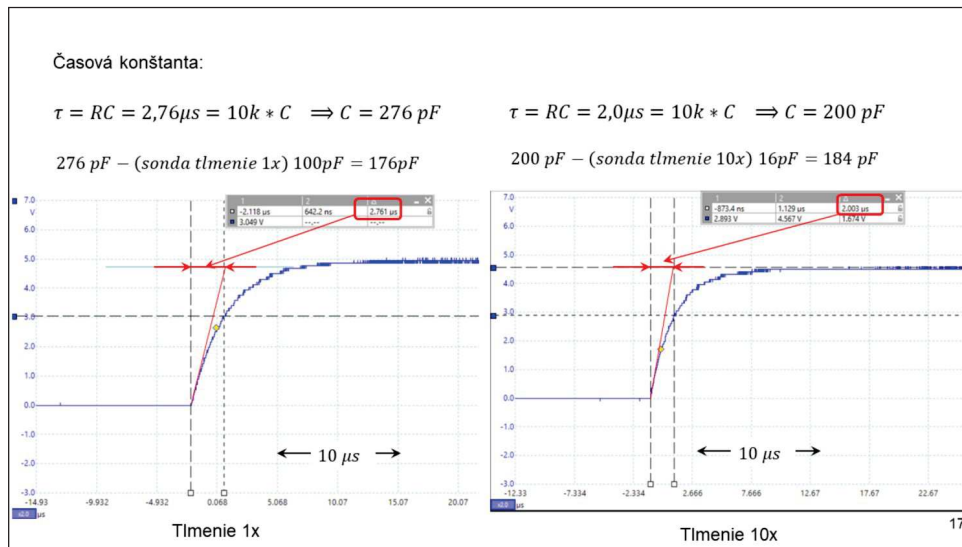


Po „uvolení tlačítka“ (rozopnutí kontaktu) sa linka opäť nabije. Ak zanedbáme vybijanie „kondenzátora“ paralelne zapojenou impedanciou osciloskopu ku kondenzátoru, môžeme z exponenciálneho priebehu nábehu napätia určiť náhradnú kapacitu linky (Samozrejme, že súčasťou je aj vstupná kapacita sondy osciloskopu (cca 100 pF)).

Trvanie prechodného procesu (nábeh/dobeh) netrvá rovnako dlho.

Na meranie som použil sondu k osciloskopu: Model P7060
Bandwidth: 60MHz; Rise Time: 5.8ns; Attenuation Ratio (AR): 1x & 10x;
Input Resistance: AR = 1x, 1MΩ±2%; AR = 10x, 10MΩ±2%
Input Capacitance: AR = 1x, 85pF~135pF ; AR = 10x, 14pF~18pF

Máme dve možnosti, buď použijeme „lepší“ osciloskop, alebo zmeníme pullup rezistor.



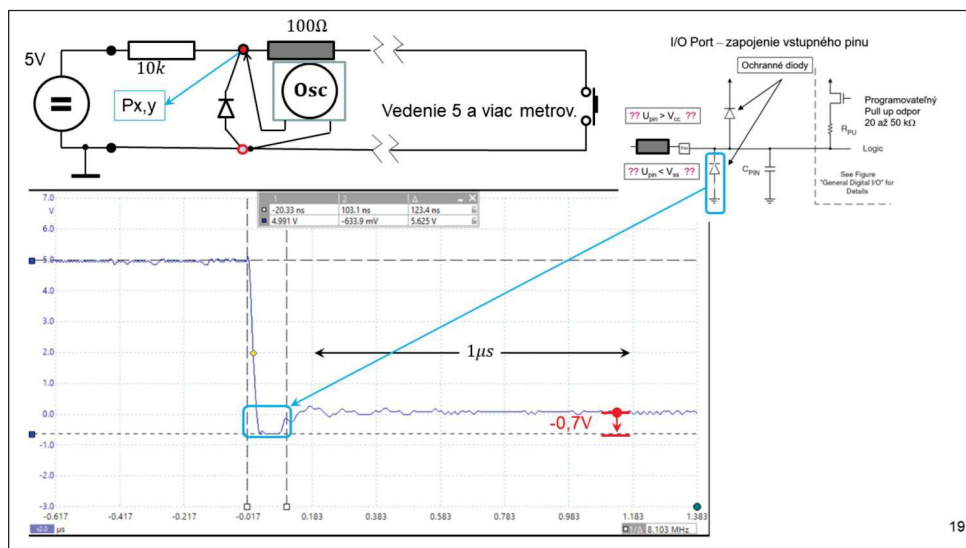
Použijeme informácie z poznámok k predchádzajúcej fólii.

Zmeníme hodnotu rezistora $R = 10 k\Omega$. Zistíme ako vplyva zmena tlmenia sondy (kapacita sondy) na prechodné procesy.

Obrázok naľavo: stredná hodnota kapacity sondy je $100 pF$.

Obrázok napravo: stredná hodnota kapacity sondy je $16 pF$.

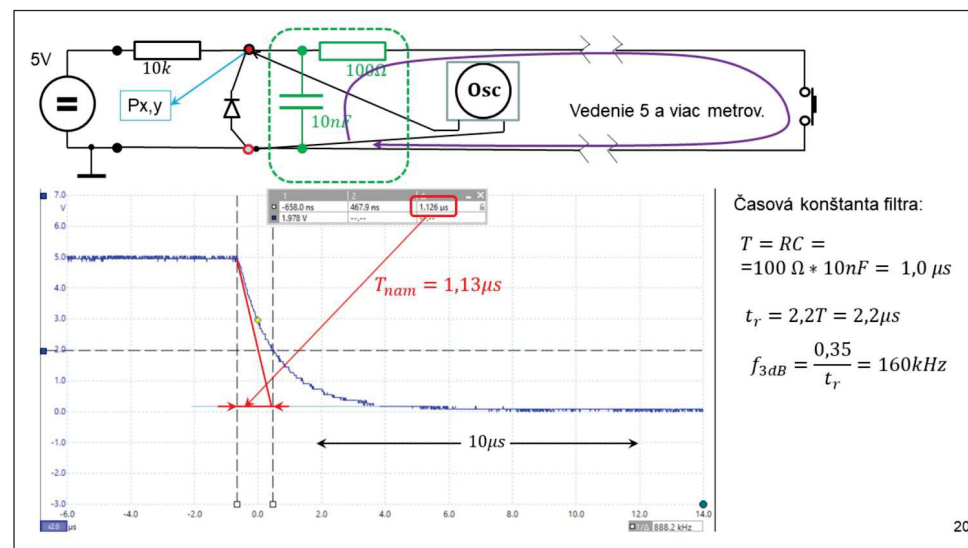
Máme tri výsledky kapacity vedenia: $164 pF$, $176 pF$ a $184 pF$. Meracím prístrojom sme odmerali $180,3 pF$. Opäť dobrá zhoda teórie a praxe.



19

Vstupné napätie $-2,5V$ je „orezané“ ochrannou diódou.

Z KL zistíme, že prúd diódou treba obmedziť. Treba zapojiť do vstupu rezistor napr. $100\ \Omega$.



20

Nepekňé priebehy prechodných procesov môžeme vylepšiť RC vstupným filtrom.

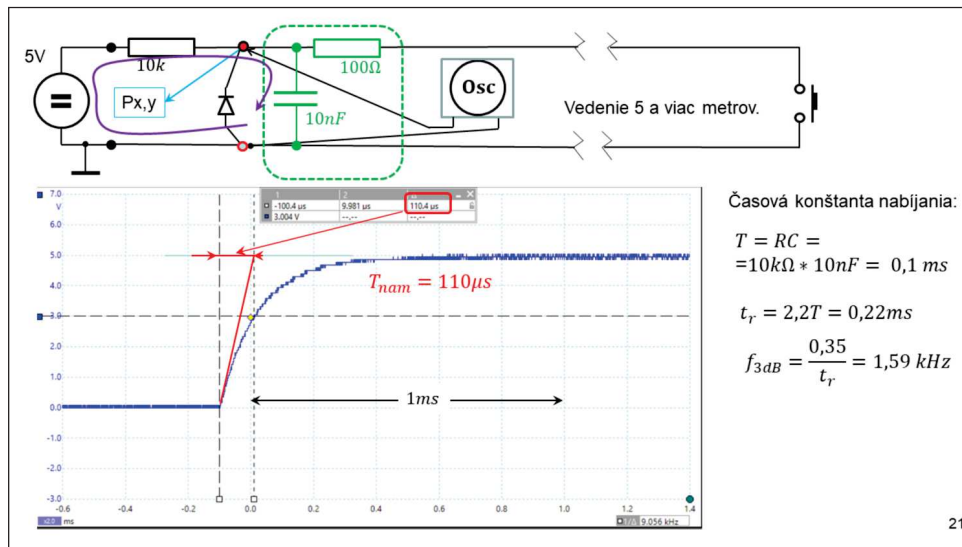
Doma nemám neobmedzený výber rezistorov a kapacitorov. Preto som nepostupoval tak, že je dané: Rise/Fall Time, resp. pásmo priepustnosti, ale vybral som rezistor $100\ \Omega$ a kapacitor $10\ nF$ a overil som namerané hodnoty.

Časová konštanta filtra:

$$T = RC = 100\ \Omega * 10nF = 1,0\ \mu s$$

$$t_r = 2,2T = 2,2\ \mu s$$

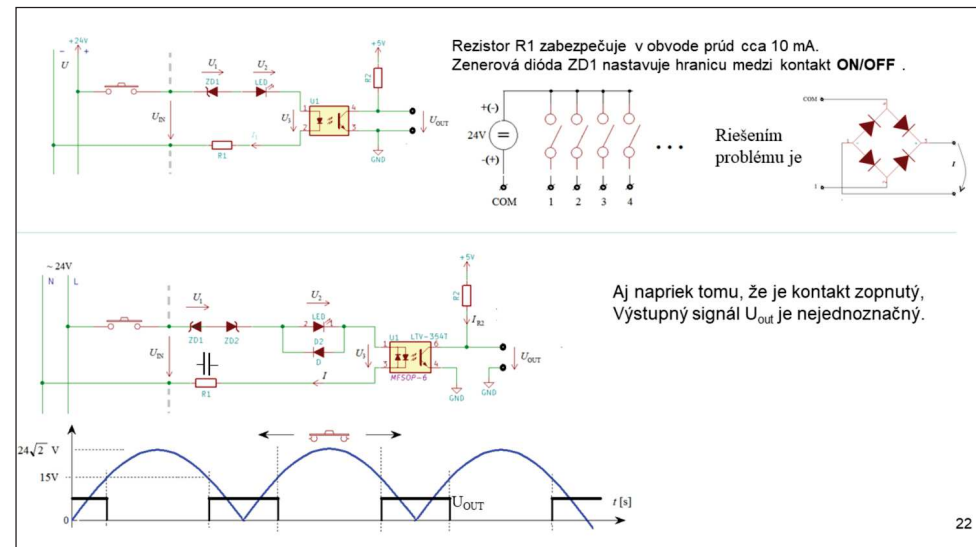
$$f_{3dB} = \frac{0,35}{t_r} = 160kHz$$



Nepekňé priebehy prechodných procesov môžeme vylepšiť RC vstupným filtrom.

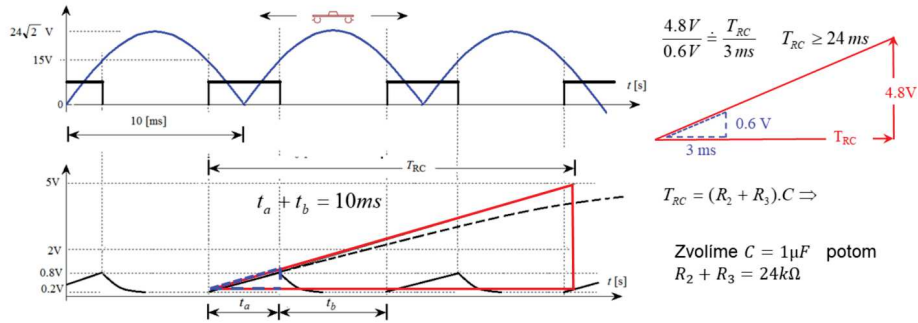
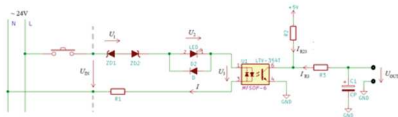
Doma nemám neobmedzený výber rezistorov a kapacitorov. Preto som nepostupoval tak, že je dané: Rise/Fall Time, resp. pásmo priepustnosti, ale vybral som rezistor 100 Ohm a kapacitor 10 nF a overil som namerané hodnoty.

Poznamenajme nábeh/dobeh netrvá rovnako dlho.



Ďalšou možnosťou ako pripojiť vzdialený kontakt je použiť optočlen. Galvanicky oddelíme obvody a naviac môžeme použiť aj iné napájanie kontaktu, dokonca aj striedavé napätie.

Príklad:
Filtrácia na výstupnej strane.
Kontakt: trvale zopnutý.



t_a vypočítame: $24\sqrt{2} \approx 34V$ zo vzťahu $\sin(2\pi f \cdot \frac{t_a}{2}) = \frac{15}{34} \Rightarrow t_a \approx 3ms$

Za čas t_a nesmie U_{OUT} opustiť „log. 0“ !!!!!

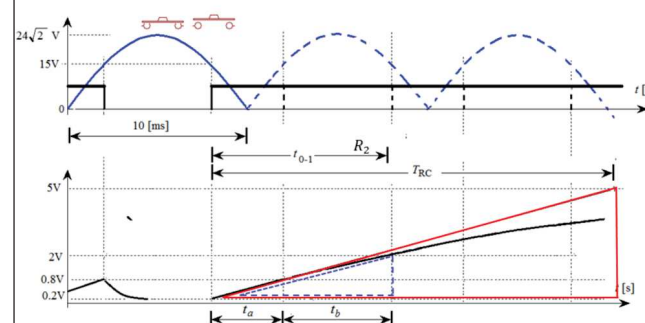
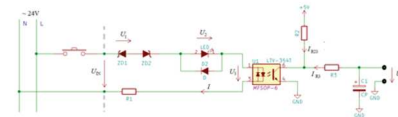
$$\frac{4.8V}{0.6V} = \frac{T_{RC}}{3ms} \quad T_{RC} \geq 24ms$$

$$T_{RC} = (R_2 + R_3) \cdot C \Rightarrow$$

Zvolíme $C = 1\mu F$ potom
 $R_2 + R_3 = 24k\Omega$

Príklad:
Kontakt rozpojíme.

Za aký čas prejde U_{OUT} zo stavu „log. 0“ do stavu „log. 1“ ?



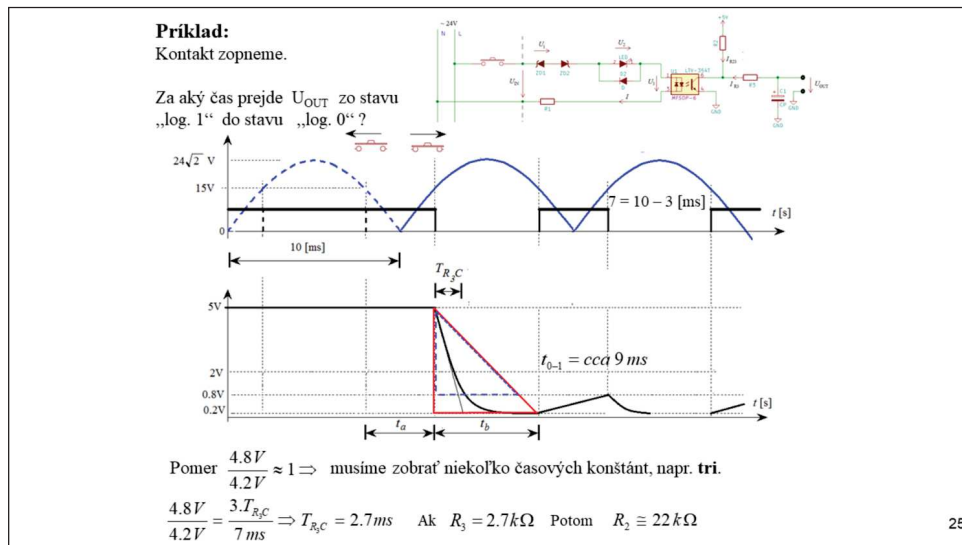
t_{0-1} vypočítame zo vzťahu: $\frac{4.8V}{1.8V} = \frac{T_{RC}}{t_{0-1}} \Rightarrow t_{0-1} = cca 9ms$

čo je menej ako jedna polperióda.

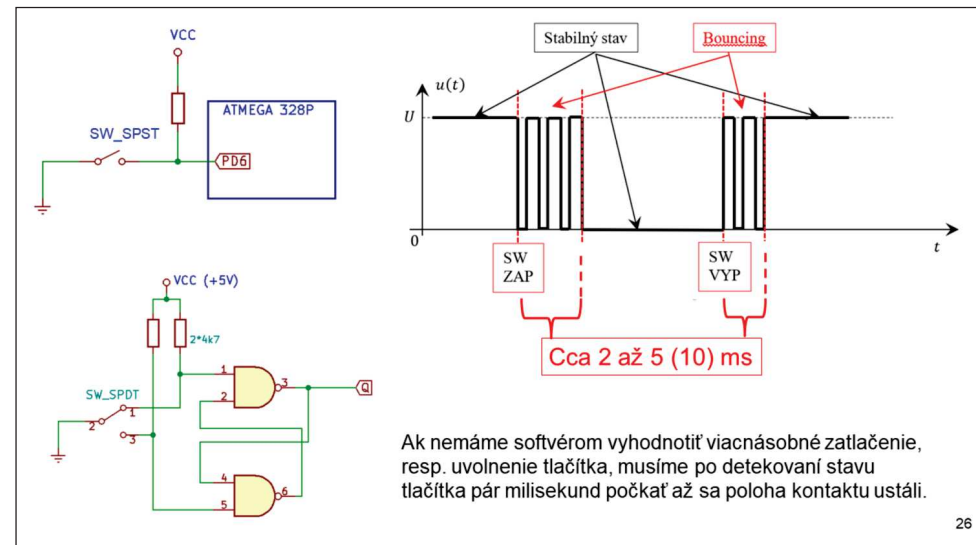
Naším cieľom, je navrhnuť filter na výstupe optočlena tak, aby reakčný čas na Zap/Vyp kontaktu bol čas jednej polperiody, t.j. 10ms.

Pri tomto riešení je to ako s 1. a 2. SV, resp. CISC a RISC. Prvé počítače nemali „nálepku“ CICS. Všetky boli tejto architektúry. Podobne to bolo v minulosti s týmto príkladom. Nemuseli sme zdôrazňovať, že v príklade sú použité úrovne TTL signálu.

Niečo počítame, niečo kontrolujem. Cieľ je aby filter „reagoval/nereagoval“ počas jednej polperiody kmitov.

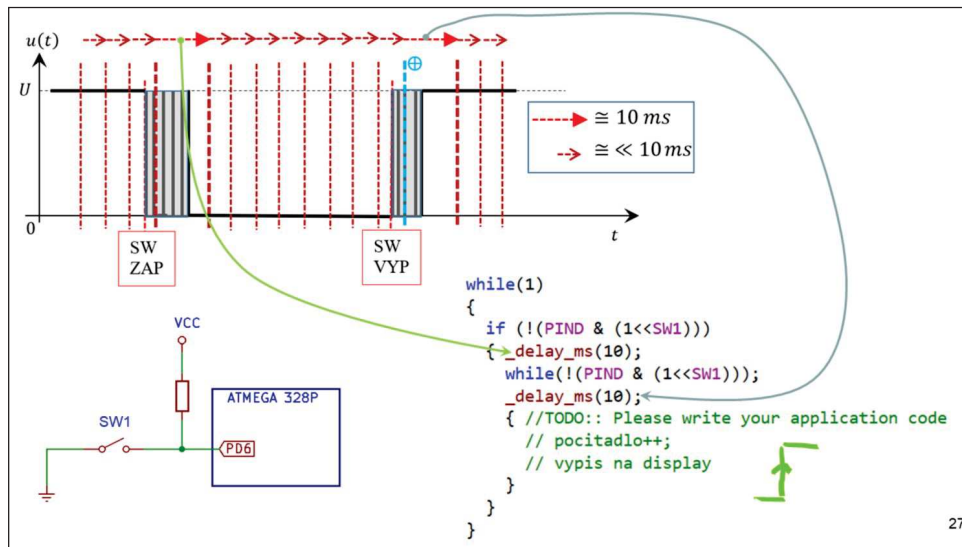


Predchádzajúci výpočet vyzerá bezchybne. Treba ale pripomenúť, že tento návrh sme realizovali v čase TTL kompatibilný. Dnes máme gro obvodov CMOS. Z toho vyplýva, zase máme čo počítať.

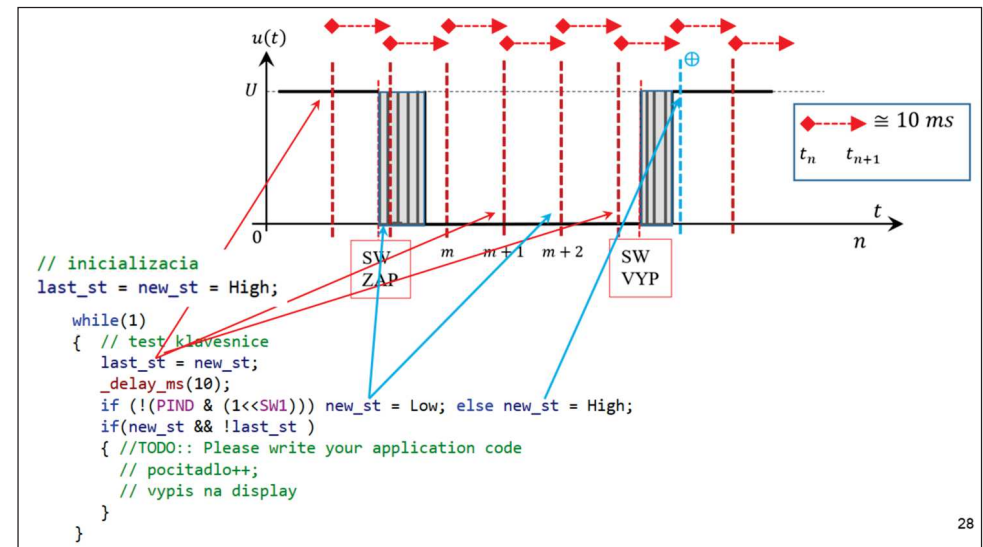


Dalo by sa povedať, že sme naznačili riešenie všetkých problémov ktoré sú spojené s „kontaktom“.

Zabudli sme na tzv. bouncing „odskakovanie“ - kmitanie kontaktu pri zopnutí, resp. pri rozopnutí. Možno si spomeniete, že na ZP sme rozprávali o RS preklápacom obvode. RS klopný obvod je takmer ideálne riešenie. Jeho nedostatkom je to, že predpokladá vypínač typu: SPDT = Single Pole, Double Throw a náš kontakt je typu SPST = Single Pole, Single Throw.



27



28

Úloha je jednoduchá. OPAKOVANE (v nekonečnej slučke) po každom zopnutí kontaktu vykonať jeden krát niečo (Například, zmeniť stav LED-ky alebo inkrementovať **pocitadlo**.) a poprípade trochu neskôr ešte niečo iné (prijat'/odoslať informáciu po sériovej linke).

Problémom úlohy s kontaktom je to, že AVR je rýchle. Ako naznačuje obr. počas stabilnej úrovne LOW by mohol program zbehnúť niekoľko krát.

Vzorkovanie, ako sa ešte len budeme učiť, sa realizuje:

- v amplitúde. Pri kontakte má len dve úrovne. Log.1 (High) a Log. 0 (Low).
- v čase. V diskretnom čase. Čas v MMP je odvodený od frekvencie oscilátora, od SC. Ešte nevieme nič o C/T a prerušovacom podsystemu tak bude generovať oneskorenie pomocou funkcie delay(x ms), resp. len vykonáme niekoľko inštrukcií. Ak "zaregistrujeme" dobežnú/nábežnú hranu, počkáme cca 10ms, aby sme nevzorkovali opakované kmitanie kontaktu.

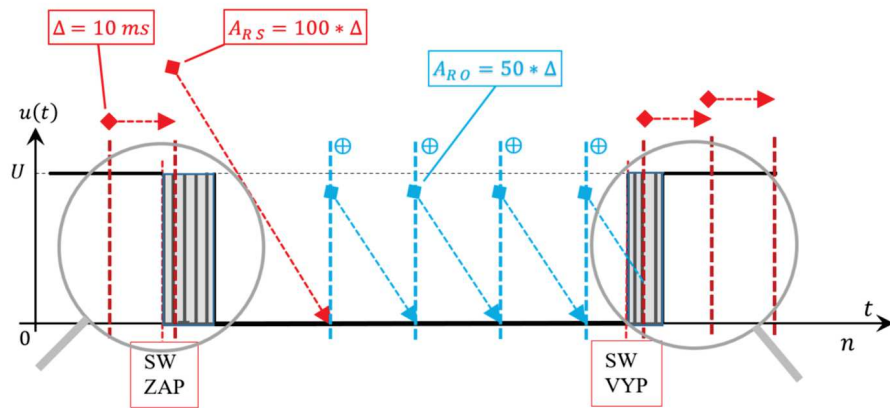
Ešte raz pripomeňme: systém reálneho času pracuje v nekonečnej slučke: while(True). Ako je zrejme z priloženého programu, tento nedokáže realizovať dve činnosti (počítač je sériovo pracujúci stroj). Ak tlačítko zatlačíme, bude do „nekonečna“ stáť.

Ďalšou nevýhodou riešenia pomocou delay() je to, že počas tejto funkcie nemôže MMP realizovať iné činnosti.

Nábežnú hranu môžeme zaregistrovať tak, že porovnáme stav kontaktu v predchádzajúcej a súčasnej vzorke. Iným spôsobom ako ošetriť kontakt proti zámkitom, je „zaregistrovať“ zmenu a potom sa po čase spýtať: Naozaj sa zmenil stav kontaktu?

Takto nejako pracuje aj prerušovací podsystem. Tá časť, kde sa vyhodnocuje nábežná, dobežná, resp. jednoducho hrana.

Autorepeat



29

Algoritmus Autorepeat funguje nasledovne: hlavná slučka zbieha každých 10 ms. Ak sa vyhodnotí prvá dobežná hrana prednastaví sa počítadlo A_{RS} na hodnotu, napr. 100, ktoré sa v hlavnej slučke každých 10ms dekrementuje. Po dopočítaní sa aktivuje rovnaká činnosť ako keby nastala nábežná hrana a prednastaví sa A_{RS} na hodnotu, napr. 50. Autorepeat sa vykoná skorej. Opäť ak dopočíta, aktivuje sa rovnaká činnosť ako keby nastala nábežná hrana. A celé sa to opakuje až do okamžiku nábežnej hrany. Teraz sa vykoná NIČ. Ak sa nábežná hrana objaví pred prvým dopočítaním počítadla, toto sa vynuluje a aktivuje sa činnosť odpovedajúca nábežnej hrane. Ak sa nábežná hrana objaví pri druhom, treťom, ... odpočítavaní, počítadlo sa vynuluje a vykoná sa NIČ.

Autorepeat:

```

// inicializacia
Int_AutRepeat = 0; // interval Autorepeat
last_st = 1; // kontakt rozopnutý
new_st = 1; // kontakt rozopnutý
auto_repeat = 0; // pocitadlo pre autorepeat

while(1)
{
    _delay_ms(10);
    // test klavesnice
    last_st = new_st;
    if(auto_repeat)auto_repeat--;
    if (!(PIND & (1<<SW1))) new_st = 0; else new_st = 1;
    if(new_st && !last_st && Int_AutRepeat)
    {
        Int_AutRepeat = 0;
        auto_repeat = 0;
        { //TODO:: Please write your application code
          // pocitadlo++;
          // vypis na display
        }
    }
    if(!new_st && last_st){auto_repeat = 100; // 100*10 = 1sek
        Int_AutRepeat = 1;
    }
    if(!auto_repeat && !new_st && !last_st ){auto_repeat = 50; // 50*10 = 0,5sek
        Int_AutRepeat = 0;
        { //TODO:: Please write your application code
          // pocitadlo++;
          // vypis na display
        }
    }
}
    
```

30

Tento program sa voči predchádzajúcim líši v nasledovnom: Je tu náznač stavového automatu a oneskorenie (čas trvania slučky) je 10ms + „niekoľko“ SC. Ak by sme 10ms –vé vzorky generovali cez prerušenie a „modro označenú časť programu“ vykonali len v čase 10 ms-vej vzorky slučka while(True) by trvala len niekoľko us alebo menej. Podľa toho či je prebeh cez slučku v čase 10ms vzorky alebo nie.

Ak budete niekedy aplikovať tento algoritmus, nezabudnite na: Bol optimalizovaný aj na počet inštrukcií aj na trvanie v čase. („Neprehadzujte poradie“.)

Digitálny potenciometer - KY-040 rotary encoder:

HT
Handson Technology

Data Sheet

Rotary Encoder for Arduino/Raspberry

The KY-040 rotary encoder is a rotary input device (as in knob) that provides an indication of how much the knob has been rotated AND what direction it is rotating in. It's a great device for stepper and servo motor control. You could also use it to control devices like digital potentiometers.

SKU: ASS-1058

Brief Data:

- Operating voltage: 5V.
- Pulses/360° Rotation: 20.
- Output: 2-bit gray code
- Mechanical Angle: 360° continuous.
- With built in push button switch (push to operate)
- Dimensions: (30 x 18 x 30) mm.
- Compatible with Arduino/Raspberry Pi controller board

www.handsontec.com

Rotary Encoder Schematic:

CLK PIN A

DT PIN B

SW

+

GND

CW Rotation →

← CCW Rotation

A to C SwRch

B to C SwRch

Open

Closed

Open

Closed

31

Ak zapojíme dva kontakty voči sebe posunuté (alebo niečo čo generuje informáciu log. 1 a log. 0) na niečo čo sa otáča, a túto informáciu vhodne spracujeme, budeme môcť odmeriavať polohu aj rýchlosť. A dokonca budeme vedieť určiť aj znamienko – teda smer.

31

Counter ++

Counter --

32

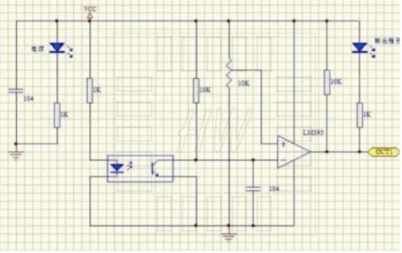
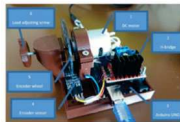

Ak sa budeme na rotačný encoder pozerat' ako na stavový automat zo 4 stavmi 10 11 01 00, dokážeme softwarovo ošetriť kmitanie kontaktov. Ale to si budeme musieť na predchádzajúcej fólii „Brief Data“ všimnúť malú nepresnosť.

Keď sa budeme snažiť priradiť stavom (binárnym číslam) 10 11 01 00 dekadické 2, 3, 1, 0 nepôjdu pekne po sebe. Na týchto binárných reťazcoch - číslach je zaujímavé to, že ich Hammingová vzdialenosť je JEDNA.

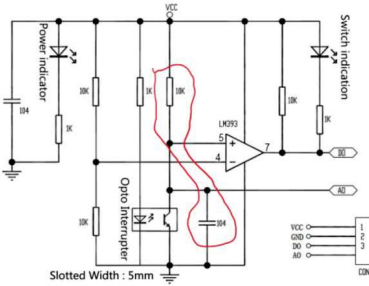
Je zaujímavé, že to čo sme robili „mechanicky“ (RC člen) potom „softwarom“ (oneskorenie) môžeme teraz urobiť len tak – použijeme vedomosti. Jednoducho povieme ono to bude fungovať správne.

Niečo podobné budeme robiť na budúcej prednáške. Prednáška bude o LCD displeji. Jedna časť sa bude venovať kontrastu displeja. Bežne sa používa na nastavenie kontrastu potenciometer. Elegantnejší spôsob je použiť PWM signál. Plnenie signálu budeme meniť pomocou tlačítok. Tento signál vyfiltrujeme pomocou RC filtra + OZ. Nakoniec vynecháme aj RC filter + OZ a PWM signál privedieme priamo na pin LCD displeja. Filtrom signálu sa „stane“ nedokonalé oko.

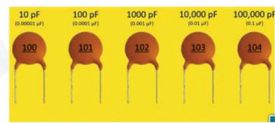
32

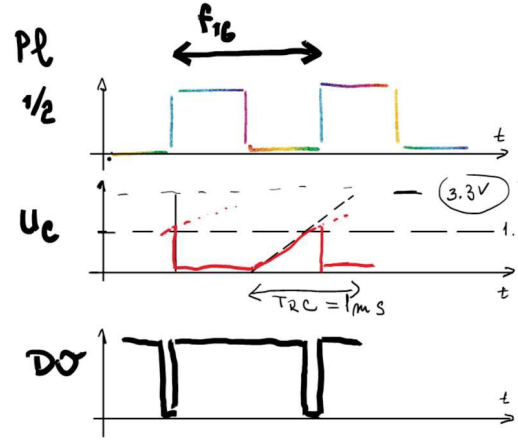
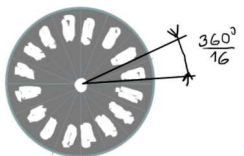
Module Circuit Diagram



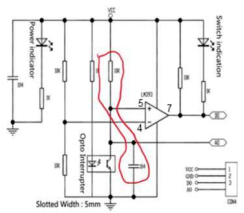
Slotted Width : 5mm



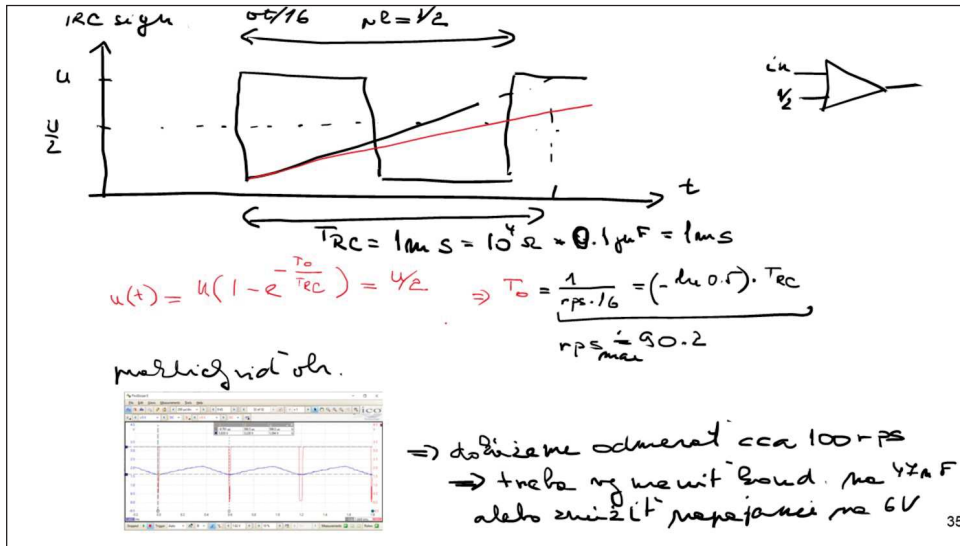
33

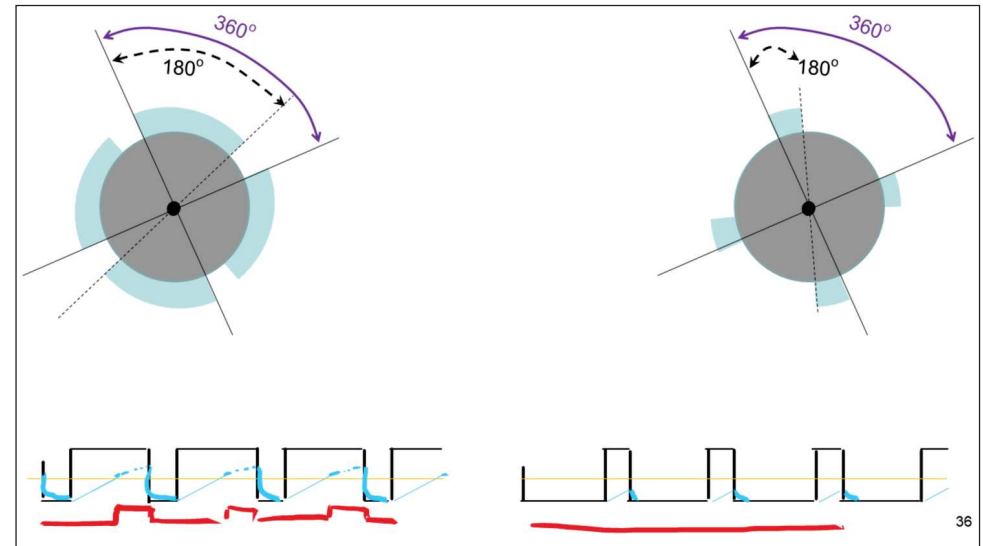
Module Circuit Diagram



34



Výsledkom je: Ak použijeme menší počet delení kruhu, môžeme merať aj väčšiu rýchlosť otáčania.



A objaví sa ďalší problém. Hmotnosť dekódovacieho kotúčika vpravo je menšia (tým aj zotrvačná hmotnosť), ale je "menej" funkčný.