

Úloha - Řídicí jednotka schodišťového automatu - čtení vstupu, ovládání výstupů

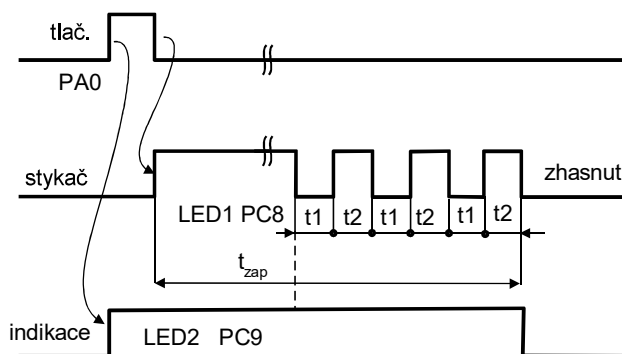
Úkol:

Vytvořte jednoduchý přístroj - časovač pro schodišťový automat s využitím STM32VL Discovery s STM32F100, který má sloužit pro ovládání světel na schodišti. Po **stisku** tlačítka na **PA0** a jeho **uvolnění** se aktivuje výstup **PC8** s **LED1**, kterým se řídí sepnutí stykače ovládajícího osvětlení na schodišti. Po doběhnutí doby t_{zap} se osvětlení definitivně zhasne. Dle předpisů musí schodišťový automat opakovaným krátkodobým zhasnutím před doběhnutím doby t_{zap} signalizovat, že se blíží definitivní zhasnutí dle obr. 1. Aktivace automatu se signalizuje pomocí **LED2** na **PC9**. Doba t_{zap} by měla být nastavitelná na dvě různé hodnoty podle délky stisku tlačítka (od stisku do uvolnění). Pokud bude tlačítko drženo dobu kratší než 1s, bude doba svitu t_{zap} krátká (např. 3s), pokud delší než 1s pak bude doba svitu dlouhá (např. 10s).

Pokud se kdykoliv **před** doběhnutím doby t_{zap} (1) opět stiskne a uvolní tlačítko (tentokrát bez ohledu na délku stisknutí), běží celá stejná doba (nastavená délkou prvního stisku) - t_{zap} (2) znovu od okamžiku uvolnění tlačítka dle obr. 2. Intervaly „blikání“ dle obr. 2 jsou $t_1=0,5$ sekundy *zhasnutí*, $t_2=0,5$ sekundy *krátké rozsvícení*.

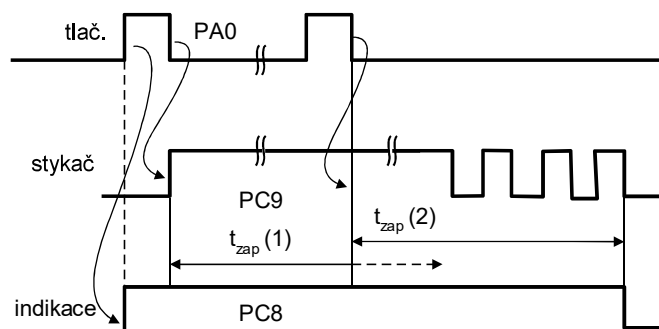
Vysvětlení:

Po stisku tlačítka na **PA0** (úroveň H - „high“, tedy log. 1) se rozsvícením **LED2** na **PC9** (přivedením úrovně H) **indikuje počátek** funkce automatu. K aktivaci stykače a rozsvícení **LED1** na **PC8** však dojde až **po uvolnění** tlačítka na **PA0**. Tento režim je u schodišťového automatu volen proto, aby nebylo možno ponechat stále sepnuté osvětlení zablokováním spínače- tlačítka na schodišti, např. zápalkou.



Obr.1 Funkce časovače po stisku tlačítka

Před koncem doběhnutí současně používané schodišťové automaty musí blikáním upozornit na blížící se okamžik definitivního zhasnutí tak, aby osoby pohybující se po schodišti byly schopny ještě za svícení dojít k nejbližšímu tlačítku a *znovu rozsvítit*, resp. *prodloužit dobu svícení*. Zde v této úloze je pro jednoduchost a rychlost odladění zvolen krátký interval $t_2 = 0,5$ sekundy. V reálném zařízení je doba t_2 delší, v řádu jednotek sekund.

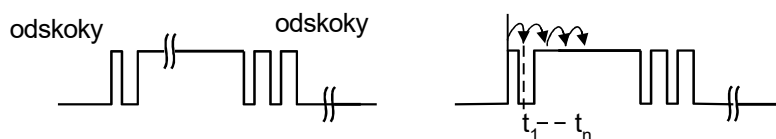


Obr.2 Funkce časovače po opakovaném stisku tlačítka

Úlohu řeší každý samostatně. V případě velkých problémů s řešením je možno – **pouze po dohodě se cvičícím** – v jednotlivých zdůvodněných případech úlohu zjednodušit na formu, kdy se jedná ovládací jednotku kuchyňského odsávače, který bude mít obdobnou funkci, avšak nebude před koncem doby krátkodobě vypínat stykač, to znamená zhasínat LED.

Problém odskoků tlačítka

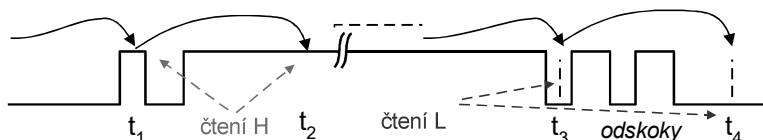
V programu by bylo vhodné programově **ošetřit** tzv. **odskoky** tlačítka dle obr. 3 (, není to však nutnou podmínkou úlohy). U některých tlačítek při stisku a při uvolnění dochází ke krátkodobé změně stavu v opačném smyslu. Tedy při stisku se kontakt v okamžiku t_1 krátkodobě přerušuje, což by mohlo být chybně interpretováno jako kompletní cyklus stisknutí a uvolnění tlačítka.



Obr. 3. Odskoky tlačítka po stisku a jeho uvolnění

Podobně při uvolňování ještě může dojít ke *krátkodobému spojení kontaktu*, který by mohl být vyhodnocen jako opětovné stisknutí tlačítka. Programové ošetření může vyjít z lidového rčení „jedna vlaštovka jaro nedělá“, tedy vyhodnocení prvního sepnutí se nebere hned jako platné, ale podmínkou je **opakované čtení stabilního stavu** tlačítka bez zjištění krátkodobých změn stavu.

Zjednodušený způsob může využít metody dvojího čtení. Pokud se prvním čtením zjistí změna stavu, dalším čtením s jistou prodlevou (např. 1 ms), za kterou by odskoky měly odeznít, se dalším čtením potvrdí skutečný stav tlačítka, tedy ve dvou následujících čtení se musí vyhodnotit stejná hodnota.



Obr.4. Čtení stavu tlačítka s potlačením vlivu odskoků zpožděným čtením

Pro tuto druhou variantu lze použít zjednodušení ve formě, kdy zpoždovací podprogram (*pprog1*) se smyčkami má dobu trvání např. 1 ms. Všechna měření času se pak mohou řešit např. v časových **elementech po 1ms**.

V této úloze by nedokonalé čtení tlačítka s odskoky (bez programového ošetření působení odskoků tlačítka) nezpůsobilo přímo nesprávnou funkci automatu, avšak došlo by k předčasnému aktivování stykače (sepnul by se již po prvním odskoku při stisku tlačítka).

Odskoky by však působily zásadní chybu v ovládacích jednotkách, kde se opakovaným stiskem tlačítka pohybuje v menu přístroje nebo se počtem stisknutí nastavují parametry. Odskoky se projevují více u starších tlačítek s kontakty postiženými zašpiněním nebo částečnou oxidací. Proto je s tímto jevem třeba počítat při návrhu programu, který s novým tlačítkem může fungovat správně, avšak po několika letech provozu může chybovat.

Poznámka k řešení čekací smyčky:

Program lze řešit různými způsoby. Zde je jeden návrh.

Program pro generování intervalu lze řešit dvoustupňově, kdy jeden podprogram (*pprog1*) řeší zpoždění 1 ms. Další, vyšší podprogram (*pprog2*) by podle vstupního parametru N_c opakovaně volal *pprog1* a po návratu z něj by vždy testoval stav tlačítka a vyhodnocoval změny jeho stavu v intervalu 1 ms. Výstup z tohoto podprogramu (ukončení) by byl dvěma způsoby.

- První způsob ukončení podprogramu *pprog2* by byl ***standardní*** po proběhnutí celého počtu N_c cyklů, tedy N_c milisekund. Výstupní informací z *pprog2* by byl aktuální ***stav tlačítka***, který byl stabilní po celou dobu trvání *pprog2*.
- Druhý způsob ukončení podprogramu *pprog2* by byl po ***zjištění změny*** stavu ***tlačítka***. Výstupní informací může být ***druh změny stavu tlačítka*** (uvolnění tlačítka, nebo stisk tlačítka) a ***počet*** skutečně ***proběhlých cyklů*** N_{SKUT}).