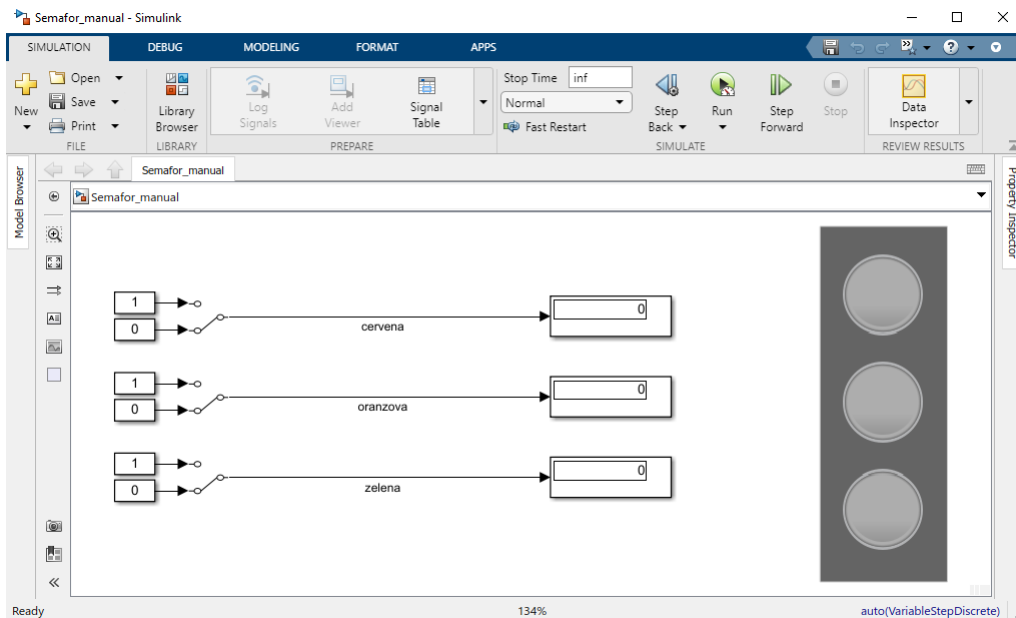
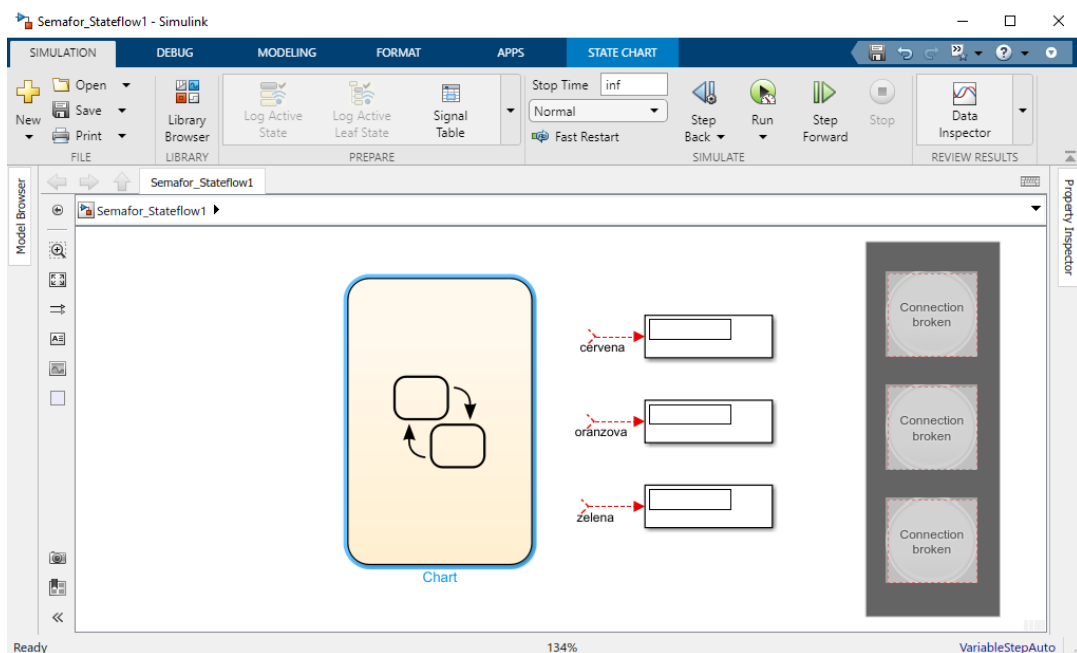


## Modelovanie udalostného systému

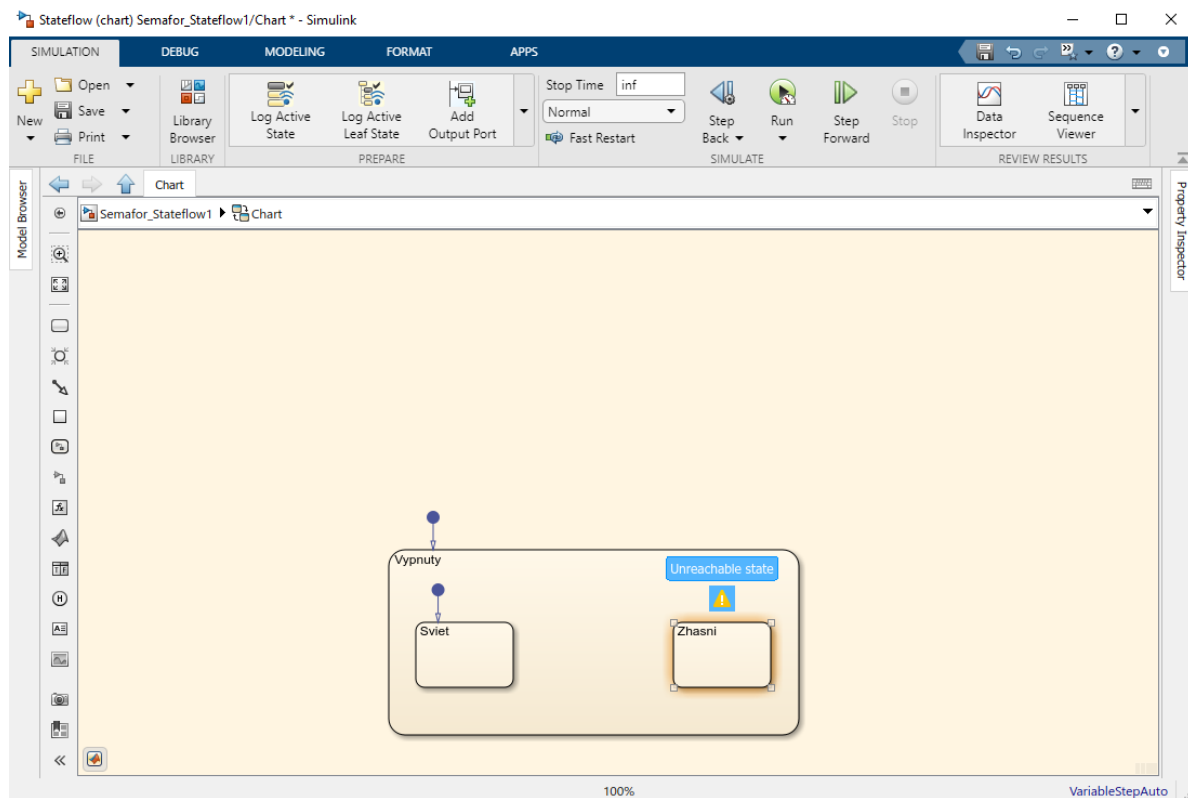
Modelovanie v prostredí Stateflow si ukážeme na jednoduchom semafore. Semafor bude mať dva stavy – vypnutý a zapnutý. Vo vypnutom stave, ktorý bude aktívny ako prvý, bude blikat oranžové svetlo. Zapnutý stav bude reprezentovať typickú prevádzku semafora, kde sa prepíname medzi jednotlivými fázami – voľno, pozor, stoj, priprav sa. Ako základ nám bude slúžiť schéma Semafor\_manual.slx.



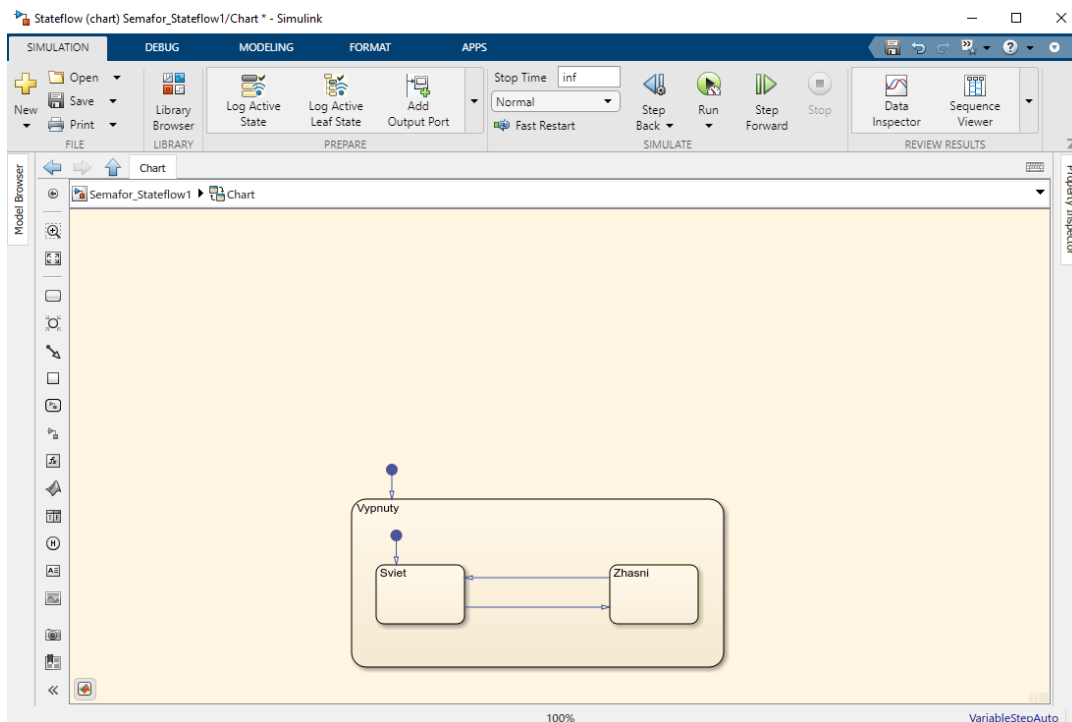
V tejto schéme si môžeme vyskúšať manuálne prepínanie svetiel na LEDkách semafora. Zmažeme bloky konštánt a prepínačov, ale signály a *Display* bloky v schéme necháme. Schému uložíme pod novým názvom, napríklad Semafor\_Stateflow1.slx. Logika udalostného systému sa do schémy pridá pomocou bloku *Chart*, ktorý nájdete v knižnici blokov v časti *Stateflow*.



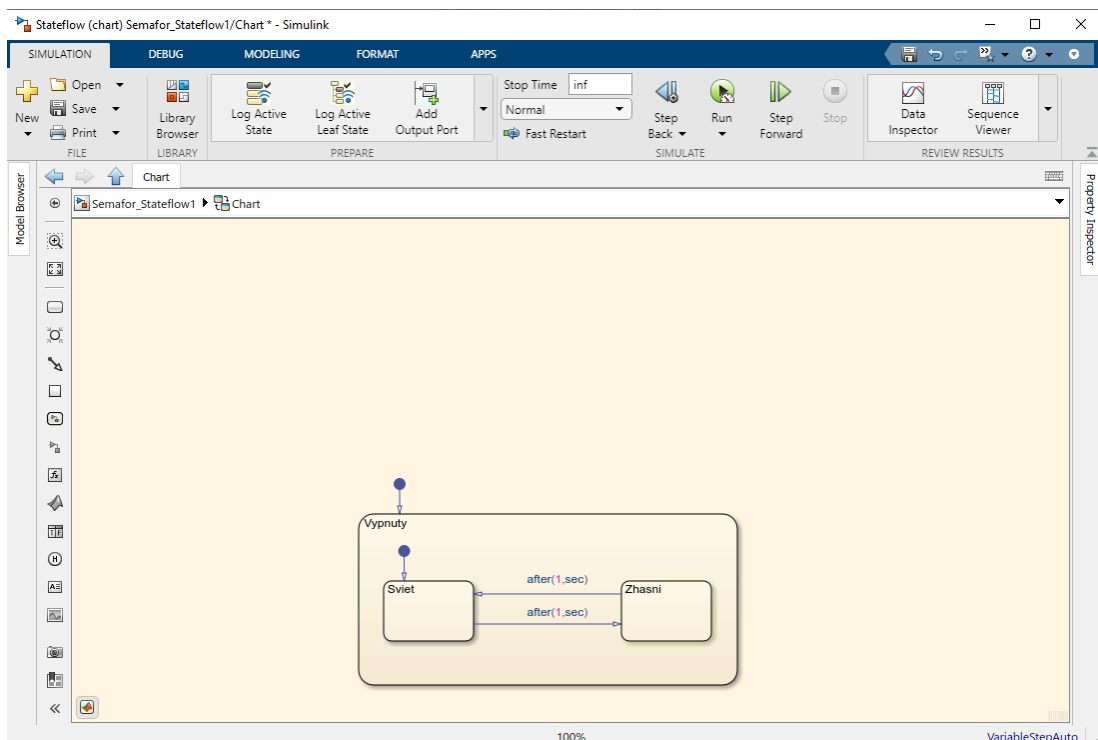
Dvoklikom na blok *Chart* sa prepne do editora, ktorý slúži na modelovanie udalostných systémov. Ľavý panel v editore obsahuje objekty, ktoré môžeme na plochu vkladať. Začneme vložení objektu *State*, ktorý reprezentuje stav. Po vložení stavu nám editor dáva možnosť pomenovať stav. Každý stav v schéme musí mať meno a tento nazveme *Vypnutý*. Naša logika môže obsahovať hierarchiu, takže stav môžeme zväčšiť potiahnutím za okraje a vložiť do stavu *Vypnutý* ďalšie dva stavy – *Sviet* a *Zhasni*.



Editor sa nám bude popri tvorbe systému snažiť radiť tak, aby sme vytvorili logiku správne. Na obrázku vidíte, že nám ohlásil varovanie oranžovým zvýraznením stavu, v ktorom sa nachádza problém. Pokiaľ by sme mali chybu, zvýraznenie by bolo červené. Varovanie naznačuje, že do stavu *Zhasni* sa nikdy nedostaneme. Aby sme sa z jedného stavu dokázali prepnúť potrebujeme medzi nimi vytvoriť prechody. Prechod vytvoríme tak, že prideme myšou na okraj stavu a pravým klikom potiahneme šípku v smere prechodu. V našom prípade sa chceme prepínať z oboch stavov, takže vytvoríme dva prechody.



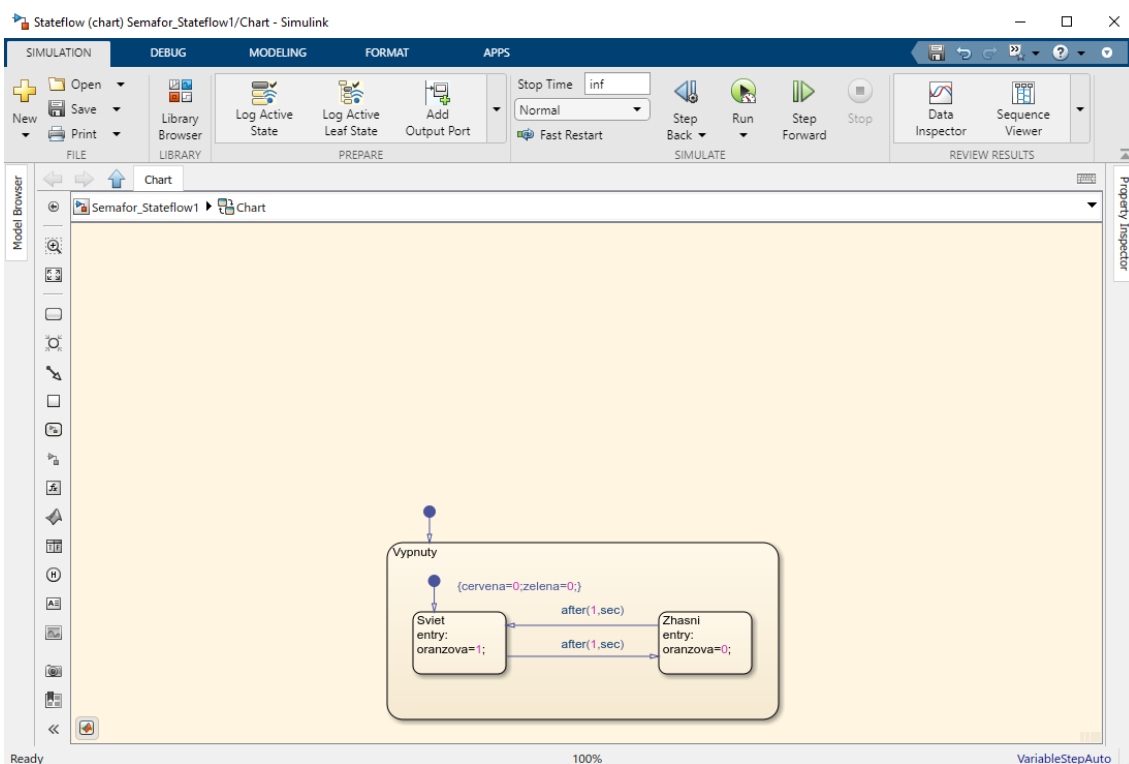
Okrem nami vytvorených prechodov schéma obsahuje ešte dva prechody, ktoré nazývame štandardné (v palete objektov ako *Default transition*). Ide o šípku s guľičkou, ktorá naznačuje, v ktorom stave začíname. Do stavu *Zhasni* sa už vieme dostať a editor nám už nehlási varovanie. Ako ale zabezpečíme, kedy sa prepne zo stavu *Sviet* do stavu *Zhasni*? Dvojklikom na prechod medzi stavmi získame možnosť editovať možnosti prechodu. Aby prechod nastal, musí nastať nejaká udalosť alebo byť splnená nejaká podmienka. V našom prípade budeme prechod zabezpečovať udalosťou, ktorá nastane po uplynutí času. Do oboch prechodov teda napíšeme *after(1, sec)*. Znamená to, že v stave počkáme jednu sekundu a následne bude možné prejsť do druhého stavu.



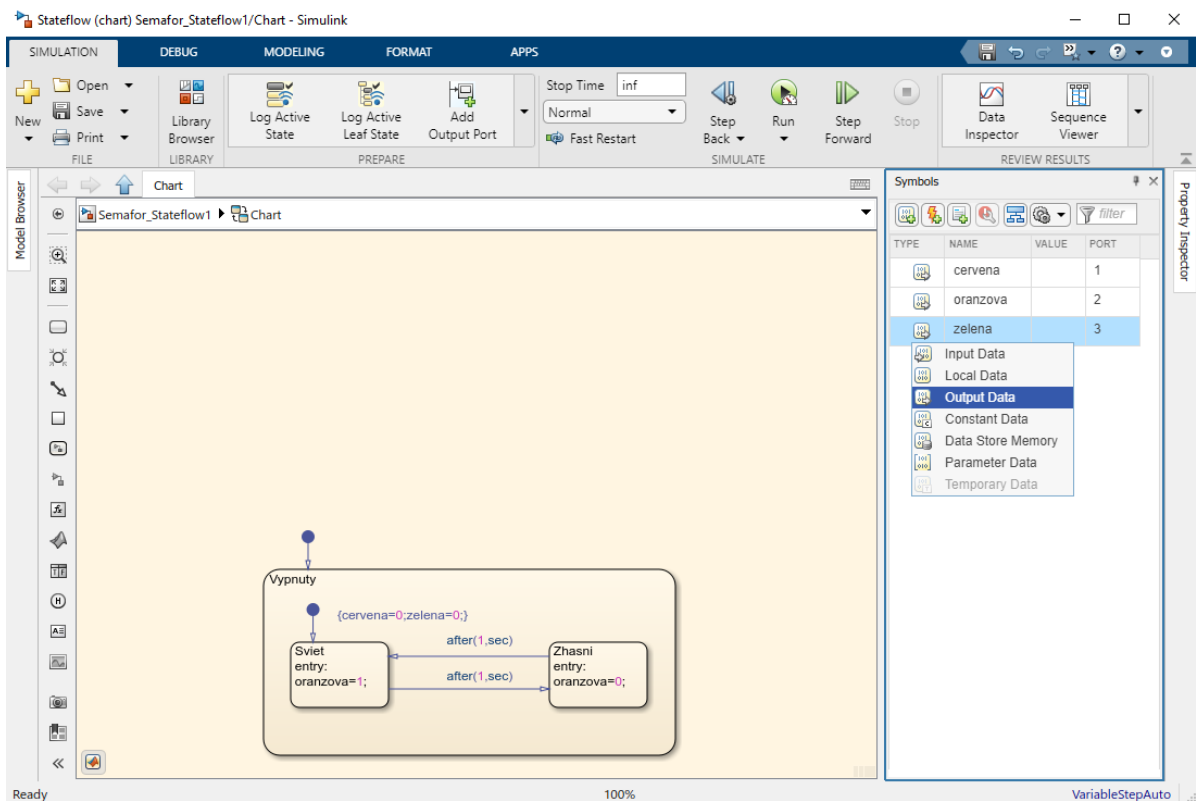
Predposledným krokom v tejto časti je definovanie rozhrania medzi prostrediami Stateflow a Simulink. Do stavov a prechodov musíme zapísať výrazy, ktoré naznačia čo sa bude zo Simulinku čítať a čo sa bude do Simulinku zapisovať. S dátami môžeme pracovať v prechodoch v zložených zátvorkách a v udalostiach stavov. Stav má tri základné udalosti:

- entry – čo sa má stať, keď do stavu prideme,
- during – čo sa má stať, keď v stave zostávame,
- exit – čo sa má stať, keď stav opúšťame.

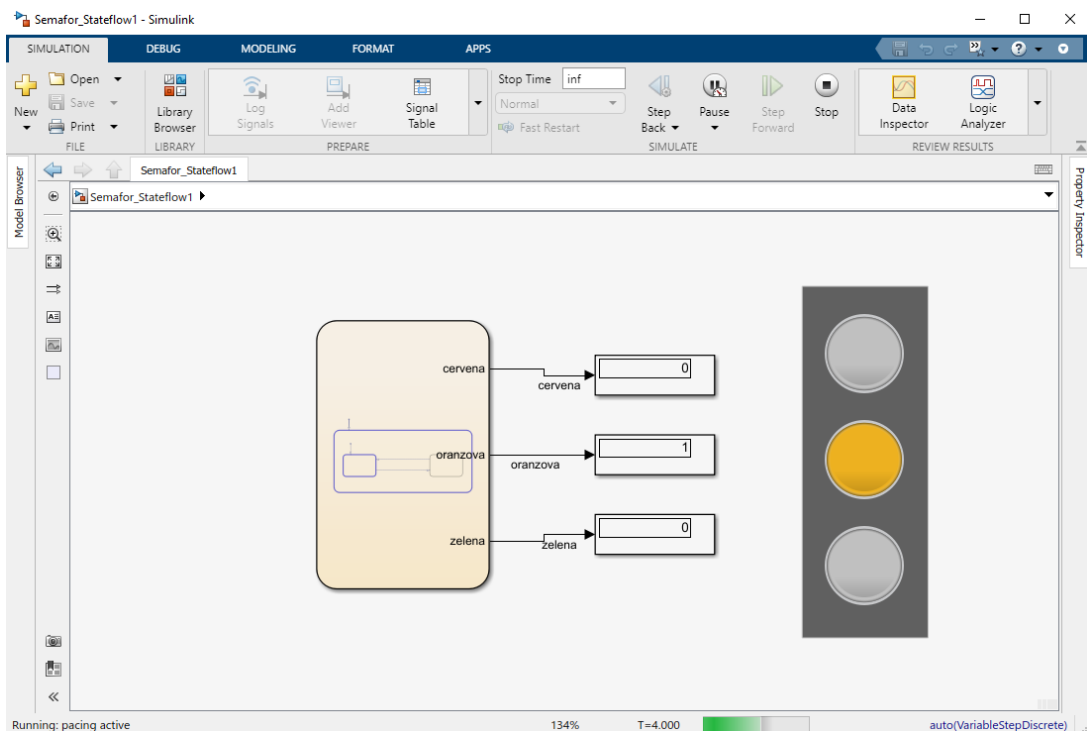
V našom príklade v štandardnom príchode do stavu *Sviet* nastavíme premenné *cervena* a *zelena* na hodnotu 0, pretože ich stačí v stave *Vypnuty* nastaviť len raz. V entry udalosti stavu *Sviet* nastavíme premennu *oranzova* na hodnotu 1 a v entry udalosti stavu *Zhasni* na hodnotu 0. Týmto dosiahneme, že pri prepínaní stavov bude svetlo oranžovej farby blikáť.



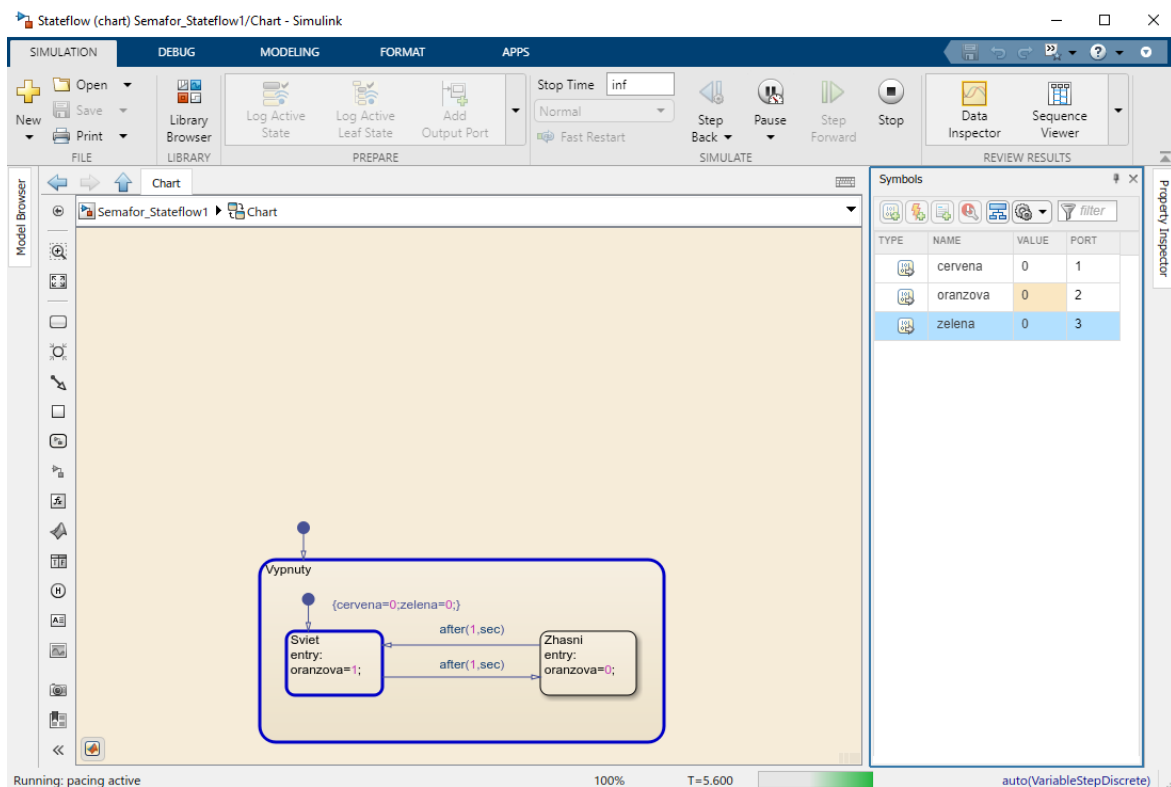
Naša logika v Stateflow síce obsahuje premenné, ale nie je definovaný ich typ. Konfiguráciu premenných môžeme urobiť pomocou panelu symbolov. V časti záložky *Simulation* v *Prepare* zapneme tlačidlom *Symbols Pane*. Môžete si všimnúť, že Stateflow vidí vytvorené premenné, avšak nemajú definovaný typ. My hodnoty premenných chceme posilať do Simulinku, takže nastavíme v stĺpčeku *Type - Output data*. Poradie výstupov určuje stĺpček *port*.



Modrou šípkou nahor, alebo pomocou mena schémy sa prepne do Simulinku. Výstupy bloku *Chart* pripojíme k našim pôvodným signálom. Pre vizualizáciu farieb budeme musieť signály naviazať na lampy semaforu. Dvakrát kliknite na lampu, po otvorení okna kliknite na signál a možnosťou *connect* označte signál. Schému máme pripravenú a simuláciu môžeme spustiť. Na semafore by sme mali vidieť blikanie oranžovej.



Čo sa však deje v bloku *Chart*? Pokiaľ sa do bloku *Chart* vrátite a spustíte simuláciu uvidíte ako rozhodovacia logika pracuje. Aktívne stavy majú zvýraznené modré okraje. Takýmto spôsobom viete vizuálne skontrolovať, či vaša logika pracuje správne.

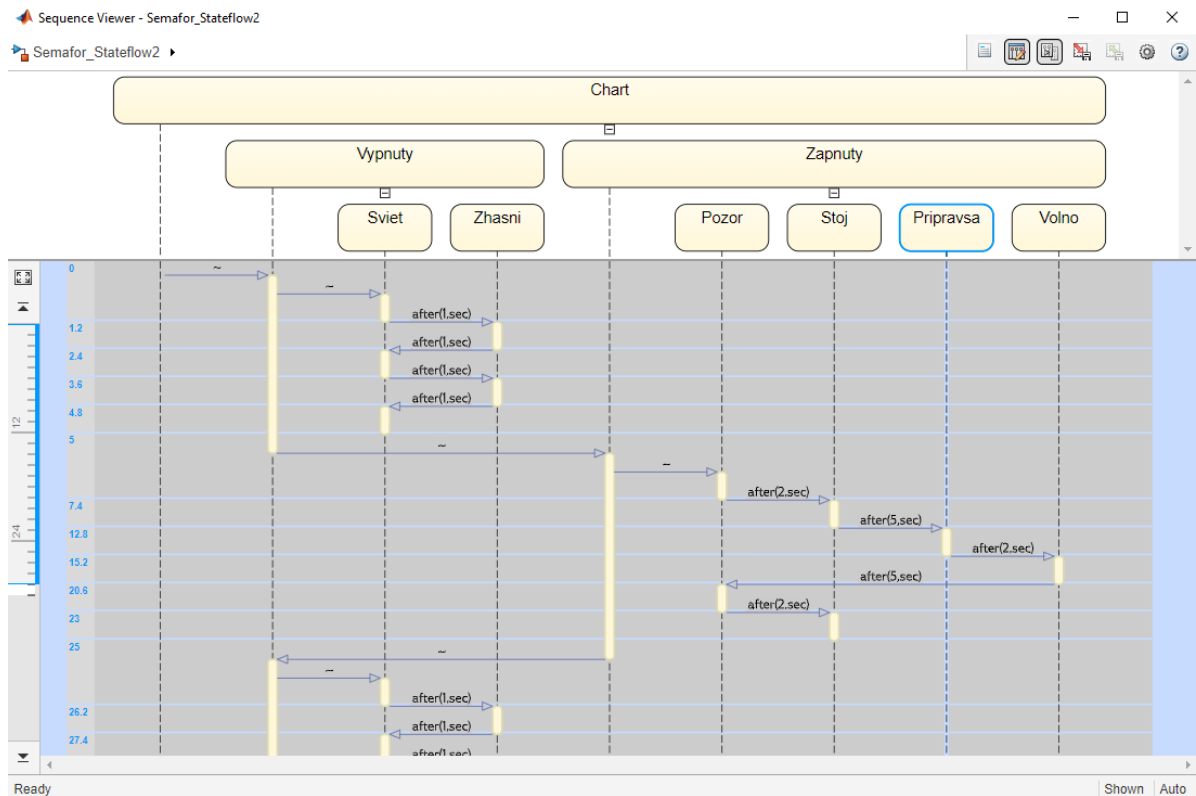


Schému si uložíme do nového súboru *Semafor Stateflow2.slx*. Do našej logiky doplníme správanie sa semafora v stave *Zapnuty*. V tomto stave začneme podstavom *Pozor*, kde bude svietiť iba oranžova. Po dvoch sekundách sa prepne do stavu *Stoj*, kde bude svietiť iba červená. Po ďalších piatich sekundách sa prepne do stavu *Pripravsá*, kde bude svietiť červená aj oranžová. Po dvoch sekundách sa prepne do stavu *Volno*, kde bude svietiť iba zelená a nakoniec po piatich sekundách sa prepne späť do stavu *Pozor*. Do logiky musíme pridať prechody medzi stavmi *Vypnuty* a *Zapnuty*. Prepínanie budeme ovládať na základe hodnoty externého signálu *ZAP*. Prechody spustí podmienka, ktorú zapisujeme v prechodoch do hranatých zátvoriek. Pokiaľ bude *ZAP* rovná nule prejdeme do stavu *Vypnuty* a pokiaľ bude *ZAP* rovná 1 prejdeme do stavu *Zapnuty*. Trúfate si stav *Zapnuty* namodelovať? Na nasledujúcom obrázku je riešenie. Nezapudnite premennú *ZAP* definovať ako *Input Data*.



## Sequence Viewer

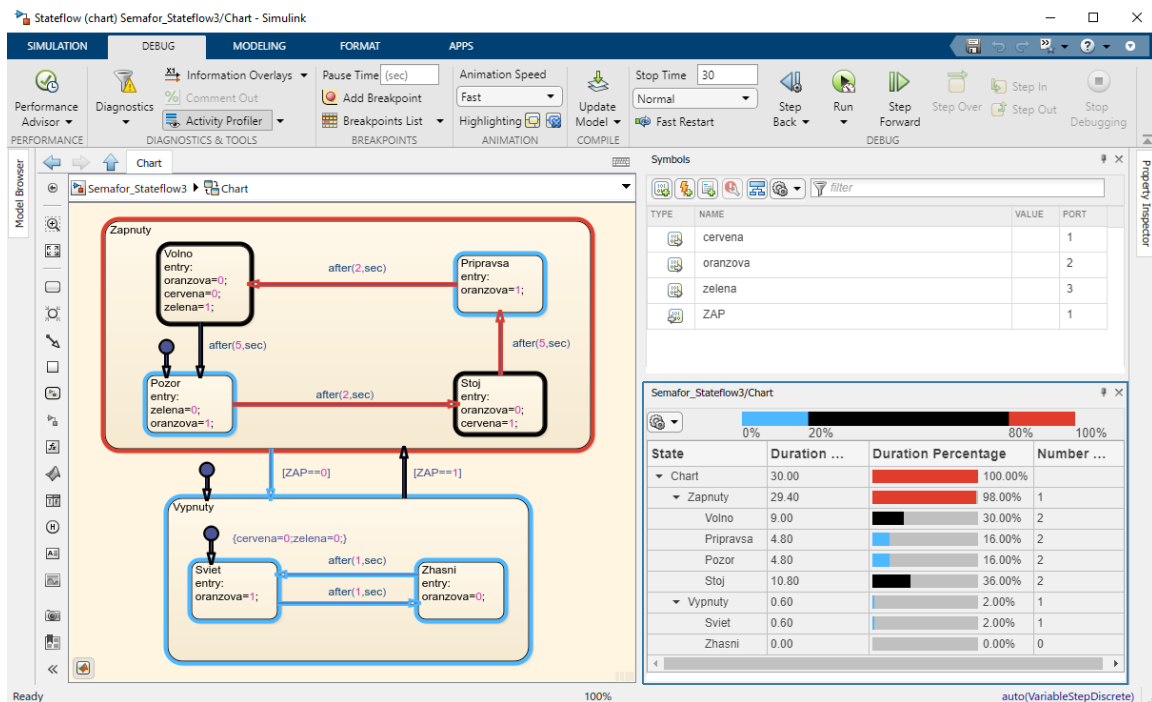
Vytvorili sme model jednoduchého semafora. Stateflow ma niekoľko nástrojov, ako náš model analyzovať. Prvým je zobrazenie prechodov počas simulácie zvýraznením aktívnych stavov, ktorý sme už videli. Druhým náhľadom je *Sequence Viewer*, ktorý nám ukáže, ako sa stavy prepínali počas simulácie. V prostredí Stateflow v záložke *Simulation* najskôr musíme zapnúť *Log Events* v časti *Prepare* a po skončení simulácie v časti *Review Results* môžeme výsledok zobraziť pomocou spomínaného *Sequence Viewera*. Pre lepšiu prehľadnosť si podstaty môžeme rozbaľiť.



## Activity Profiler

Druhým nástrojom analýzy je od verzie R2020b *Activity Profiler*. Pomocou tohto nástroja vieme pozorovať ako často boli zapnuté stavy, prípadne do ktorých stavov sme sa nedostali. Pred tým ako vyskúšame ako *Activity Profiler* funguje si schému pre istotu uložíme pod novým menom *Semafor Stateflow3.slx*. V Simulinku môžeme nahradiť pulzný generátor konštantou s hodnotou 1. To nám zabezpečí, že sa do niektorých stavov nedostaneme. Prepne sa do záložky *Debug* a zapneme *Activity Profiler*. Model odsimulujeme a objaví sa nám nové okno. V jeho nastaveniach môžeme zvoliť zvýraznenie stavov a prechodov (*State highlighting*, *Transition highlighting*) a *Color Scheme* prepne na *HotCold*. V prostredí Stateflow uvidíme, v ktorých prechodoch a stavoch sme trávili najviac času. Vidíme, že semafor bol viac zapnutý ako vypnutý.





## Záver

Na jednoduchšom príklade semafora sme si ukázali modelovanie udalostného systému v prostredí Stateflow a možnosti jeho analýz. Pokiaľ vás Stateflow zaujal môžete si toto prostredie vyskúšať pomocou bezplatnej TRIAL licencie na 30 dní. Pokiaľ by ste sa chceli o modelovaní udalostných systémov dozvedieť viac môžete si urobiť bezplatný online kurz zameraný na Stateflow, prípadne Simulink alebo MATLAB na MATLAB Academy:

<https://matlabacademy.mathworks.com/>