

## 1.2.1. Valós idejű rendszerek

Sok eszközt, az egyszerű háztartási gépektől kezdve az ipari gyártósorokig számítógép vezérel. Ezek a számítógépek közvetlen kapcsolatban állnak a hardverrel. A vezérlő- szoftver ezekben az eszközökben beágyazott valós idejű szoftver<sup>1</sup>, amelynek a hardver által generált eseményekre a megfelelő vezérlőjelekkel kell válaszolnia. Jellemző hogy ennek a hardverrendszerbe beágyazott vezérlőszoftvernek valós időben kell reagálnia a beérkező eseményekre.

DEF.: „A *valós idejű rendszer* olyan szoftverrendszer, amelyben a rendszer helyes működése egyrészt a rendszer által adott eredményektől, másrészt attól az időtől függ, amennyi idő alatt ez az eredmény előáll.”

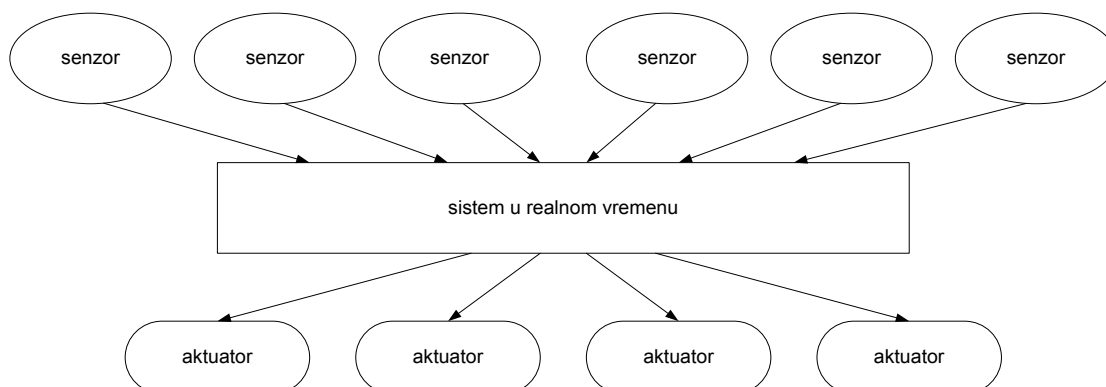
A valós idejű szoftverrendszerek különböznek a többi szoftverrendszerektől. Működésük akkor megfelelő ha a beérkező eseményekre az előírt határidőn belül reagálnak. A beágyazott rendszerek rövidebb vagy hosszabb határidőn belül tudnak reagálni, de nem mindegyik tud folyamatosan a megadott határértéknél rövidebb idő alatt reagálni, vagyis nem minden beágyazott rendszer, valós idejű rendszer. A valós idejű rendszereket inger/válasz rendszernek lehet tekinteni.

A valós idejű beágyazott rendszerek viselkedését leírhatjuk:

- a lehetséges ingerekkel/stimulációkkal,
- az erre adott lehetséges válaszokkal,
- és a válaszok előállításához szükséges idővel.

Az ingerek két csoportba sorolhatóak:

- *periodikus stimuláció* – jellemző rájuk, hogy az ingerek egyenletes időközönként jelentkeznek. Például: egy időzítő hatására a rendszer minden 25 miliszekundumban leolvassa egy kapcsoló állapotát (inger) és az állapot függvényében valamilyen műveletet hajt végre (válasz).
- *aperiodikus stimuláció* – jellemző rájuk, hogy az ingerek előre nem látható időpontokban jelentkeznek. Ezt a típusú stimulációt a beágyazott rendszerek leggyakrabban megszakításokon (###EN## interrupt) keresztül kezelik le. Például: megszakítási kérelem jelzi, hogy a beolvasási puffer megfelelő mennyiségű adatot tartalmaz, vagyis megkezdődhet az adatok kiolvasása.



\$\$\$ PLG KEP-KEP-KEP újrarajzolni \$\$\$

1.ábra – a valós idejű rendszer egyszerűsített modellje

<sup>1</sup> embedded realtime software – beágyazott, valós idejű szoftver

Periodikus stimulációs jeleket a valós idejű rendszerekben a leggyakrabban olyan érzékelők (##EN## sensor) szoktak előállítani, amelyek közvetlen kapcsolatban vannak a rendszerrel és amelyek feladata a környezet állapotáról való információgyűjtés. A rendszer erre adott válaszai a hardvert működtető elemekhez (##EN## actuator – működtető szerkezet) jutnak el, így befolyásolván a környezet állapotát.

Az aperiodikus stimulációs jelek származhatnak az érzékelőktől, az aktuátoroktól vagy magától a felhasználótól. Ezek a jelek általában valamilyen rendkívüli eseményt vagy állapotot jeleznek mint pl.: a működési mód megváltozása vagy hardveres hiba, amelyekre a rendszernek reagálnia kell. Az 1. ábrán van ábrázolva a valós idejű rendszerek *érzékelő – valós idejű rendszer – működtető elem* modellje.

A valós idejű rendszernek minden stimulációra tudnia kell válaszolnia, annak ellenére hogy ezek az ingerek különböző időpontokban jelentkeznek. Ennél fogva olyan architektúrája kell hogy legyen, hogy a válaszként adott vezérlőjel mihamarabb eljusson a célba. Ez szekvenciális programokkal nem lehetséges. Ezért a valós idejű szoftverrendszereket, együttműködő konkurens folyamatokként tervezik meg. Az ilyen folyamatok lekezelésére a valós idejű rendszerek leggyakrabban valós idejű operációs rendszert (##EN## Real Time Operating System - RTOS) tartalmaznak/alkalmazzanak. Az ilyen operációs rendszerek lehetőségeit magasabb szintű programozási nyelvekből tudjuk kihasználni, rendszerhívásokon keresztül.

A valós idejű rendszer inger-válasz egyszerűsített modelljéből eljuthatunk egy általános architektúráis modellhez, amely három típusú folyamatot foglal magában (2. ábra):

- minden érzékelőtípus rendelkezik egy folyamattal amely (le)kezeli az érzékelőt;
- a feldolgozó folyamat számításokat végez a beérkező adatok alapján;
- a vezérlő folyamat vezéri a működtető elemeket.



\$\$\$ PLG KEP-KEP-KEP újrarajzolni \$\$\$

2. ábra. – három típusú folyamatot tartalmazó általános architektúráis modell

Ez az architektúráis modell alapján lehetséges az adatok gyors begyűjtése az érzékelőkről, még mielőtt új adatok jelennének meg, valamint lehetőséget biztosít arra, hogy az adatok feldolgozása és a válaszkció egy későbbi időpontban legyenek elvégezve. Ebből az általános architektúráis modelltől két gyakorlati architektúra vezethető le:

- Felügyelő- és vezérlő rendszer;
- adatgyűjtő rendszer.