# Přerušovací podsystém mcu

* =Softwarová nebo hardwarová událost (změna stavu), která může vyvolat dočasné zastavení programu
* Softwarové testování: opakované testování příznaku (plýtvá softwarovým časem, nelze určit čas mezi provedením události a zachycením; tzv. „pooling“)
* Hardwarové události: změna hodnoty na I/O pinu, přetečení čítače. dokončení převodu u AD převodníku
* Softwarové přerušení: Neplatná adresa, neplatný operační znak

# Obsluha

* Podprogram, který se spustí při vzniku přerušení (pokud je povoleno)
* Přeruší se běžící program
* Řízení se předá na nějakou speciální adresu (na této adrese je buď přímé program obsluhy, nebo další adresa/skok na skutečné umístění podprogramu)
* Této adrese se říká „vektor přerušení“
* Vektor přerušení je trojího druhu:
  + Vyhrazený (každá událost má vlastní interrupt vector)
  + Smíšený (některé události mají vyhrazený, některé ho sdílí)
  + Společný (pouze jeden interrupt vector => umí skákat na rlzné adresy, které se softwarově vyhodnocují; pomalejší, ale umožňuje definování priority)

# Povolení

* Každá událost má vlastní povolovací bit (po resetu jsou všechny přerušení zakázaná)
* Musí být povolen Global Interrupt Enable (používá se, pokud nechceme, aby obsluha přerušení byla přerušena jinou obsluhou)

# Příznak

* Informace, že došlo k příslušné události (nezávisle na povolení)
* Nastavuje se na jedničku, pokud vzniklo dané přerušení
* Softwarově vynulovat kdykoli
* Některé systému umí hardwarově vynulovat automaticky po spuštění obsluhy
* Pokud by se flag nenuloval, volala by se obsluha pořád dokola

# Priorita

* Probíhající obsluha přerušení může být přerušena událostí s vyšší prioritou
* Některé systémy mají registry, ve kterých lze každému interruptu nastavit prioritu (pořád ale lze nastavit u některých přerušení stejnou prioritu)
* U ATmega64 pořadí vykonání obsluhy přerušení zároveň určuje vektor přerušení (nižší adresa = vyšší priorita)
* U systémů s jedním interrupt vectorem se priorita definuje softwarově

# Kontext

* Po přerušení se program dostane na určitou adresu
* Na této adrese se musí na přerušení softwarově reagovat (probíhají instrukce, které mají vliv na obsah registrů)
* => po dokončení obsluhy zůstanou registry modifikovány
* Obsluha přerušení musí zajistit, aby se obsah registrů po ukončení obsluha rovnal obsahu před spuštěním obsluhy
* Na začátku se uloží obsah registrů, jejichž obsah se bude modifikovat (např. na zásobník) a před návratem z přerušení se původní obsah (tzv. kontext) obnoví

# Zpracování

* Během každého instrukčního cyklu se kontroluje, jestli nedošlo k přerušení
* Pokud ano a je povolen=, tak se obsah následující instrukce uloží na zásobník a do PC (program counter) se nahraje příslušný vektor přerušení
* To zajístí skok na adresy obsluhy přerušení a zakáže se Global Interrupt Enable
* Provede se obsluha přerušení
* Povolí se Global Interrupt Enable
* Ze zásobníku se obnoví adresa poslední instrukce, kde program před přerušením byl
* Tato adresa se nahraje do program counteru
* Provede se minimálně jedna následující instrukce, během které se opět zkontrolují příznaky přerušení
* Pokud instrukce trvá více než jeden instrukční cyklus, tak se před přerušením musí dokončit