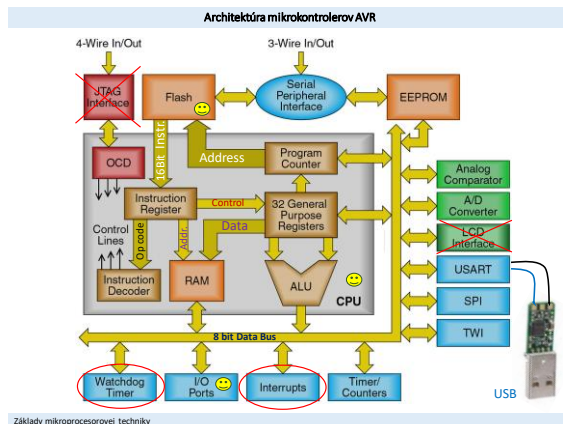


Týždeň	Prednášky	Cvičenia
1.	Úvod do predmetu- Číselné systémy v elektronike; kódovane čísel- Jazyk C- Úvod do digitálnej elektroniky.	Instalácia: IDE- Microchip-Studio -driver pre Arduino- kit
2.	Poľovodičové pamäte	Práca s Microchip-Studio, definovanie projektu
3.	ATmega328, pamäť, generovanie hodinových signálov, manažment napájania, reset	Práca s Microchip-Studio preklad, simulácia projektu
4.	Vstupy-výstupy	Práca so vstupmi a výstupmi
5.	Externé a interné prerušenia, Watchdog časovač	Práca s prerušeniami Výber zadania na zápočet
6.	Sériové komunikačné prostriedky USART – Tx	Práca s USART – TX. Práca na zadani
7.	Sériové komunikačné prostriedky USART – Rx	Práca s USART – RX. Práca na zadani
8.	Časovače – počítadlá	Práca s TCO. Práca na zadani
9.	Časovače – počítadlá	Práca s TCO. Práca na zadani
10.	A/D prevodník	Programovanie A/D. Práca na zadani
11.	EEPROM, FUSE bity	Programovanie EEPROM. Práca na zadani
12.	I2C zbernica	Zápočet - Praktická práca s MCU
13.	Predtermín	Zápočet - Praktická práca s MCU

Základy mikroprocesorovej techniky

1



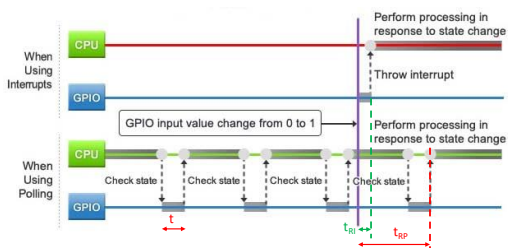
Základy mikroprocesorovej techniky

2

Obsluha požiadaviek externých zariadení MCU

Obsluhu požiadaviek externého zariadenia v počítačovom systéme je možné hardvérovú realizovať dvoma spôsobmi:

- procesor nepretržite kontroluje stavov I/O vývodov (zariadení), či je potrebná ich obsluha (Polling). Je to jednoduché, ktoré neefektívne využíva čas procesora.
- spracovanie (hardvérové) požiadavky o obsluhu zariadenia procesorom cez prerušenie činnosti procesora. **Prerušenie** činnosti procesora je metóda na efektívnu asynchrónnu obsluhu podnetov.



Základy mikroprocesorovej techniky

3

ATmega328P – Externé a Interné prerušenia MCU (16.)

- Po vyvolaní prerušenia sa potrebné dáta ukladajú do **registrov** a do **zásobníkovej pamäti**, aby sa použili pre návrat k pôvodnej činnosti procesora.
- Jednotlivé prerušenia majú priradenú **prioritu**.
- Vykonávaná obsluha prerušenia môže byť prerušená len prerušením s vyššou prioritou.
- Vektory prerušení sú uložené v pamäti programu v **tabuľke vektorov** a obsahujú adresy obsluhujúcich programov pre jednotlivé prerušenia.
- Prerušenie s **najvyššou prioritou** je **RESET**.
- Prerušenia majú priradené **enable bits**, ktorými je možné zakázať/povoľiť prerušenie.
- Súčasne musí byť povolený aj bit **„I“** (Global Interrupt Enable) v **Status registri SREG**.
- Externé prerušenia **INT0**, **INT1**, **PCINT0**, **PCINT1**, **PCINT2** môžu byť aktivované hrana signálu, nízkou úrovňou alebo zmenou úrovne.

ISR (TIMER_OVF_vect) {

Number	Program Address	Source	Interrupt Definition
1	0x0001	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0003	INT1	External Interrupt Request 1
4	0x0004	PCINT0	Pin Change Interrupt Request 0
5	0x0005	PCINT1	Pin Change Interrupt Request 1
6	0x0006	PCINT2	Pin Change Interrupt Request 2
7	0x0007	NOT	Working Time-out Interrupt
8	0x0008	TIMERO_COMPAR	Timer/Counter 0 Compare Match A
9	0x0009	TIMERO_COMPAR	Timer/Counter 0 Compare Match B
10	0x000A	TIMERO_OVF	Timer/Counter 0 Overflow
11	0x000B	TIMERO_CAPT	Timer/Counter 0 Capture Event
12	0x000C	TIMERO_COMPAR	Timer/Counter 1 Compare Match A
13	0x000D	TIMERO_COMPAR	Timer/Counter 1 Compare Match B
14	0x000E	TIMERO_OVF	Timer/Counter 1 Overflow
15	0x000F	TIMERO_OVF	Timer/Counter 1 Overflow
16	0x0010	TIMERO_COMPAR	Timer/Counter 2 Compare Match A
17	0x0011	TIMERO_OVF	Timer/Counter 2 Overflow
18	0x0012	SPI_STC	SPI Serial Transfer Complete
19	0x0013	USART_RX	USART Rx Complete
20	0x0014	USART_TX	USART Tx Complete
21	0x0015	USART_RX	USART Rx Complete
22	0x0016	ADC_COMP	ADC Conversion Complete
23	0x0017	EE_READY	EEPROM Ready
24	0x0018	ANALOG_COMP	Analog Comparator
25	0x0019	TWI	2-wire Serial Interface
26	0x001A	SPI_READY	SPI Serial Transfer Complete

Základy mikroprocesorovej techniky

4

ATmega328P – Externé a Interné prerušenia MCU (16.)

Number	Program Address	Source	Interrupt Definition
1	0x0001	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0003	INT1	External Interrupt Request 1
4	0x0004	PCINT0	Pin Change Interrupt Request 0
5	0x0005	PCINT1	Pin Change Interrupt Request 1
6	0x0006	PCINT2	Pin Change Interrupt Request 2
7	0x0007	NOT	Working Time-out Interrupt
8	0x0008	TIMERO_COMPAR	Timer/Counter 0 Compare Match A
9	0x0009	TIMERO_COMPAR	Timer/Counter 0 Compare Match B
10	0x000A	TIMERO_OVF	Timer/Counter 0 Overflow
11	0x000B	TIMERO_CAPT	Timer/Counter 0 Capture Event
12	0x000C	TIMERO_COMPAR	Timer/Counter 1 Compare Match A
13	0x000D	TIMERO_COMPAR	Timer/Counter 1 Compare Match B
14	0x000E	TIMERO_OVF	Timer/Counter 1 Overflow
15	0x000F	TIMERO_OVF	Timer/Counter 1 Overflow
16	0x0010	TIMERO_COMPAR	Timer/Counter 2 Compare Match A
17	0x0011	TIMERO_OVF	Timer/Counter 2 Overflow
18	0x0012	SPI_STC	SPI Serial Transfer Complete
19	0x0013	USART_RX	USART Rx Complete
20	0x0014	USART_TX	USART Tx Complete
21	0x0015	USART_RX	USART Rx Complete
22	0x0016	ADC_COMP	ADC Conversion Complete
23	0x0017	EE_READY	EEPROM Ready
24	0x0018	ANALOG_COMP	Analog Comparator
25	0x0019	TWI	2-wire Serial Interface
26	0x001A	SPI_READY	SPI Serial Transfer Complete

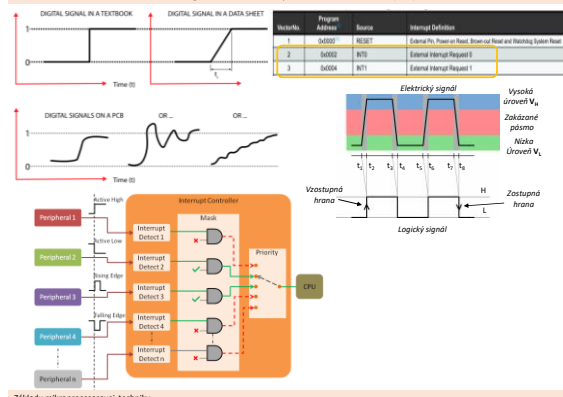
The most typical and general program setup for the Reset and Interrupt Vector Addresses is:

Address	Label	Code	Comments
0x0000	0x0000	0x0000	RESET
0x0001	0x0001	0x0001	INT0
0x0002	0x0002	0x0002	INT1
0x0003	0x0003	0x0003	PCINT0
0x0004	0x0004	0x0004	PCINT1
0x0005	0x0005	0x0005	PCINT2
0x0006	0x0006	0x0006	PCINT2
0x0007	0x0007	0x0007	WATCHDOG_TIMEOUT
0x0008	0x0008	0x0008	TIMERO_COMPAR
0x0009	0x0009	0x0009	TIMERO_COMPAR
0x000A	0x000A	0x000A	TIMERO_OVF
0x000B	0x000B	0x000B	TIMERO_CAPT
0x000C	0x000C	0x000C	TIMERO_COMPAR
0x000D	0x000D	0x000D	TIMERO_COMPAR
0x000E	0x000E	0x000E	TIMERO_OVF
0x000F	0x000F	0x000F	TIMERO_OVF
0x0010	0x0010	0x0010	TIMERO_COMPAR
0x0011	0x0011	0x0011	TIMERO_OVF
0x0012	0x0012	0x0012	TIMERO_OVF
0x0013	0x0013	0x0013	TIMERO_OVF
0x0014	0x0014	0x0014	TIMERO_OVF
0x0015	0x0015	0x0015	TIMERO_OVF
0x0016	0x0016	0x0016	TIMERO_OVF
0x0017	0x0017	0x0017	TIMERO_OVF
0x0018	0x0018	0x0018	TIMERO_OVF
0x0019	0x0019	0x0019	TIMERO_OVF
0x001A	0x001A	0x001A	TIMERO_OVF
0x001B	0x001B	0x001B	TIMERO_OVF
0x001C	0x001C	0x001C	TIMERO_OVF
0x001D	0x001D	0x001D	TIMERO_OVF
0x001E	0x001E	0x001E	TIMERO_OVF
0x001F	0x001F	0x001F	TIMERO_OVF
0x0020	0x0020	0x0020	TIMERO_OVF
0x0021	0x0021	0x0021	TIMERO_OVF
0x0022	0x0022	0x0022	TIMERO_OVF
0x0023	0x0023	0x0023	TIMERO_OVF
0x0024	0x0024	0x0024	TIMERO_OVF
0x0025	0x0025	0x0025	TIMERO_OVF
0x0026	0x0026	0x0026	TIMERO_OVF
0x0027	0x0027	0x0027	TIMERO_OVF
0x0028	0x0028	0x0028	TIMERO_OVF
0x0029	0x0029	0x0029	TIMERO_OVF
0x002A	0x002A	0x002A	TIMERO_OVF
0x002B	0x002B	0x002B	TIMERO_OVF
0x002C	0x002C	0x002C	TIMERO_OVF
0x002D	0x002D	0x002D	TIMERO_OVF
0x002E	0x002E	0x002E	TIMERO_OVF
0x002F	0x002F	0x002F	TIMERO_OVF
0x0030	0x0030	0x0030	TIMERO_OVF
0x0031	0x0031	0x0031	TIMERO_OVF
0x0032	0x0032	0x0032	TIMERO_OVF
0x0033	0x0033	0x0033	TIMERO_OVF
0x0034	0x0034	0x0034	TIMERO_OVF
0x0035	0x0035	0x0035	TIMERO_OVF
0x0036	0x0036	0x0036	TIMERO_OVF
0x0037	0x0037	0x0037	TIMERO_OVF
0x0038	0x0038	0x0038	TIMERO_OVF
0x0039	0x0039	0x0039	TIMERO_OVF
0x003A	0x003A	0x003A	TIMERO_OVF
0x003B	0x003B	0x003B	TIMERO_OVF
0x003C	0x003C	0x003C	TIMERO_OVF
0x003D	0x003D	0x003D	TIMERO_OVF
0x003E	0x003E	0x003E	TIMERO_OVF
0x003F	0x003F	0x003F	TIMERO_OVF

Základy mikroprocesorovej techniky

5

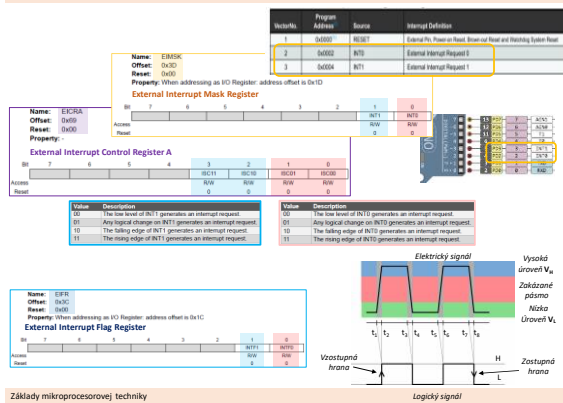
ATmega328P – externé prerušenia INT0, INT1 (17.)



Základy mikroprocesorovej techniky

6

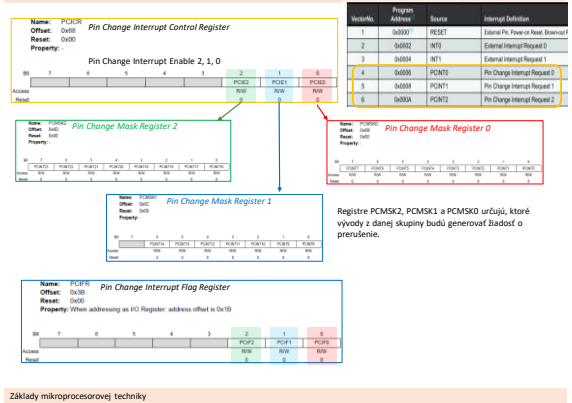
ATmega328P – externé prerušenia INT0, INT1 (17.)



Základy mikroprocesorovej techniky

7

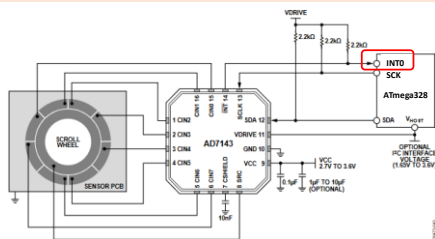
ATmega328P – externé prerušenia PCINT0, PCINT1, PCINT2 (Pin Change Interrupt) (17.)



Základy mikroprocesorovej techniky

8

ATmega328P – príklad použitia externého prerušenia



- MCU reaguje na požiadavku prerušenia (INT0) a načíta informácie z AD7143 o aktivovanom kapacitnom senzore cez I2C zbernicu (SCX, SCL) informáciu.
- Použitím prerušenia nemusí MCU pravidelne vyčítavať informácie o stave jednotlivých senzorov v určitých časových intervaloch (*Polling*). Tento čas môže venovať inej činnosti. Informácia o aktivovanom senzore sa načítava iba v prípade, ak je aspoň jeden senzor aktívny.
- Pooling* (alebo *polling*) je technika, pri ktorej procesor pravidelne v určitých časových intervaloch kontroluje stav zariadenia.

Základy mikroprocesorovej techniky

9

ATmega328P – externé prerušenia INT0, INT1

```
int.c
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/interrupt.h>

//ISR (INT0_vect) //Obsluha externého prerušenia INT0
ISR (INT0_vect) {
    PORTB |= (1<<PORTB5); _delay_ms(100); PORTB &= ~(1<<PORTB5);
    EIFR |= 1<<INTF0;
}

//ISR (INT1_vect) //Obsluha externého prerušenia INT1
ISR (INT1_vect) {
    PORTB |= (1<<PORTB5); _delay_ms(1000); PORTB &= ~(1<<PORTB5);
    EIFR |= 1<<INTF1;
}

void EXTINT_init(void) {
    EIMSK |= 1<<INT0 | 1<<INT1; //Enable INT0, INT1
    EICRA |= 1<<ESR0 | 1<<ESR1; //Trigger on falling edge
    //EICRA |= 1<<ISC0 | 1<<ISC1; //Trigger on rising edge
    EIFR |= 1<<INTF0 | 1<<INTF1;
    PORTD |= (1<<PORTD0) | (1<<PORTD1); //Pull-up on INT0, INT1
}

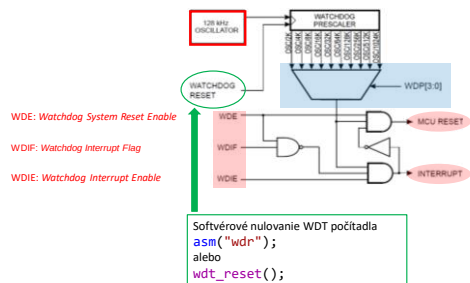
int main(void) {
    DDRC |= (1<<DDRB5); //Nastaví vývod LED na výstup
    EXTINT_init(); //Inicializuje INT0, INT1
    sei(); //Povolí globálne prerušenia
    while(1) {}
}
```

Základy mikroprocesorovej techniky

10

ATmega328P – watchdog časovač (15.8, 15.9)

- WDT (Watchdog Timer) je počítač, ktoré po pretečení generuje : *interný reset, interrupt* alebo oboje.
- Ak je WDT povolený, je potrebné ho *programovo nulovať*.
- Watchdog oscilátor (128kHz) sa povoľuje alebo zakazuje pomocou FUSE bitu WDTON (štandardne povolený).
- Časový interval počítača sa nastavuje pomocou bitov WDTCR.WDP[3:0].

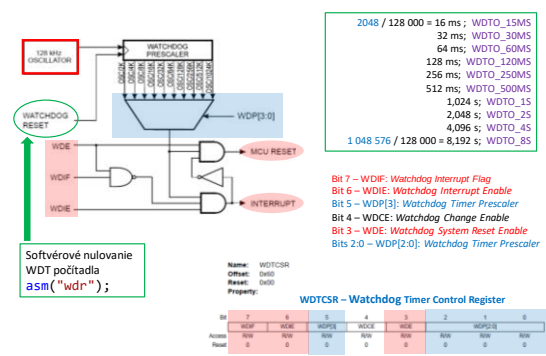


Základy mikroprocesorovej techniky

11

ATmega328P – watchdog časovač (15.8, 15.9)

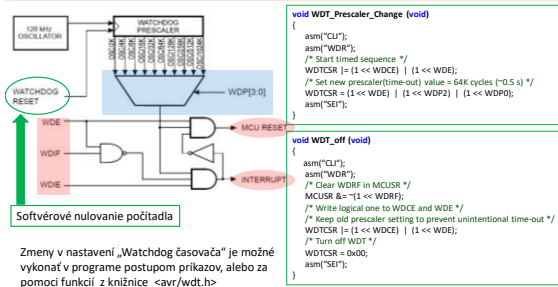
- Časový interval počítača sa nastavuje pomocou bitov WDTCR.WDP[3:0].



Základy mikroprocesorovej techniky

12

ATmega328P – watchdog časovač (15.8, 15.9)



http://nongnu.org/avr-libc/user-manual/group_avr_watchdog.html

```
wdt_disable(); //zakáže watchdog reset
wdt_enable(WDTO_500MS); //povolí watchdog reset a nastaví WDT časovač na 500 ms
```

Základy mikroprocesorovej techniky

13

ATmega328P – watchdog časovač (15.8, 15.9)



http://nongnu.org/avr-libc/user-manual/group_avr_watchdog.html

```
wdt_disable(); //zakáže watchdog reset
wdt_enable(WDTO_500MS); //povolí watchdog reset a nastaví WDT časovač na 500 ms
```

Základy mikroprocesorovej techniky

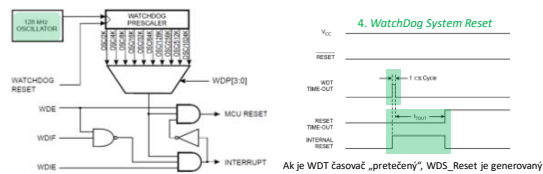
14

ATmega328P – WDS_reset (15.6)

Po uplynutí časového limitu Watchdog časovača sa vygeneruje krátky resetovací impulz v trvaní jedného hodinového cyklu. Zostupná hrana tohto impulzu spustí časové oneskorenie t_{WDTR} .

Typ. Time-out ($V_{\text{CC}} = 5.0\text{V}$)	Typ. Time-out ($V_{\text{CC}} = 3.0\text{V}$)	Number of Cycles
0ms	0ms	0
4.1ms	4.3ms	512
65ms	60ms	8K (8,192)

$$8192 / 128000 \text{ Hz} = 0,064 \text{ s}$$



Základy mikroprocesorovej techniky

15

ATmega328P – WDS_reset (15.6)

Softvérové nulova

Zmeny v nastavení „Watchdog časovača“ je možné vykonať v programe postupom príkazov, alebo za pomoci funkcií z

Základy mikroprocesorovej techniky

16