

1. Przeczytaj:

<https://forbot.pl/forum/topic/5180-kurs-programowanie-mikrokontrolerow-avr-w-jezyku-assembler-czesc-4/>

2. Napisz szablon kodu assembler i zacznij od części programu która poprawnie ustawi ADC - po ludzku wrzucasz jedynki i 0 w te bity rejestrów ADMUX oraz ADCSRA tak żeby się włączyło z daną szybkością etc.

Przykład:

ADC_Init:	
ldi r16,3	; Select ADC3
out ADMUX, r16	; Enable ADC, Single Mode conversion
ldi r16, 10000101b	; ADC Interrupt disable, Prescaler division factor = 32
out ADCSR,r16	; this gives an ADC clock frequency of 4e6/32=125kHz.
sbi ADCSR,ADSC	; Start conversion
 Wait:	
sbis ADCSR,ADIF	; Wait until the conversion is completed
rjmp Wait:	
 in r16,ADCL	
in r17,ADCH	; Place ADCH in r16:r17.

Rysunek 1:

3. Napisz kod który przeniesie dane z bufora ADC (kieszonka gdzie znajdują się kolejne pomiary napięcia z odpowiedniej nóżki mikokontrolera) na dany port (8 nóżek - bajt)

Przykład:

main: //program glowny

sbic ADCSRA, ADIF //jeśli bit ADIF w rejestrze ADCSRA nie jest ustawiony(konwersja nie gotowa) to pomiń następną instrukcję

rcall load_OCR //wywołaj poprogram który załaduje wartość z przetwornika ADC do rejestru OCR0

rjmp main //instrukcja zostanie wykonana gdy konwersja nie jest jeszcze gotowa ,tym samym pętla ta wykonuje się, gdy oczekujemy na konwersję

load_OCR: //podprogram ładujący wartość z rejestru ADCH do rejestru OCR0

in R16, ADCH //załaduj wartość z ADCH do R16

out OCR0, R16 //załaduj wartość z R16 do OCR0

sbi ADCSRA, ADSC //ustaw bit ADSC w rejestrze ADCSRA , powoduje to wyzwolenie kolejnej konwersji ADC

ret //powrót do podprogramu main