

студент: ..... група:..... преподавател: .....

## АНАЛОГОВ ИНТЕРФЕЙС НА ЕМК НС11

### I. ТЕОРЕТИЧНА ПОСТАНОВКА. ЕКСПЕРИМЕНТИ

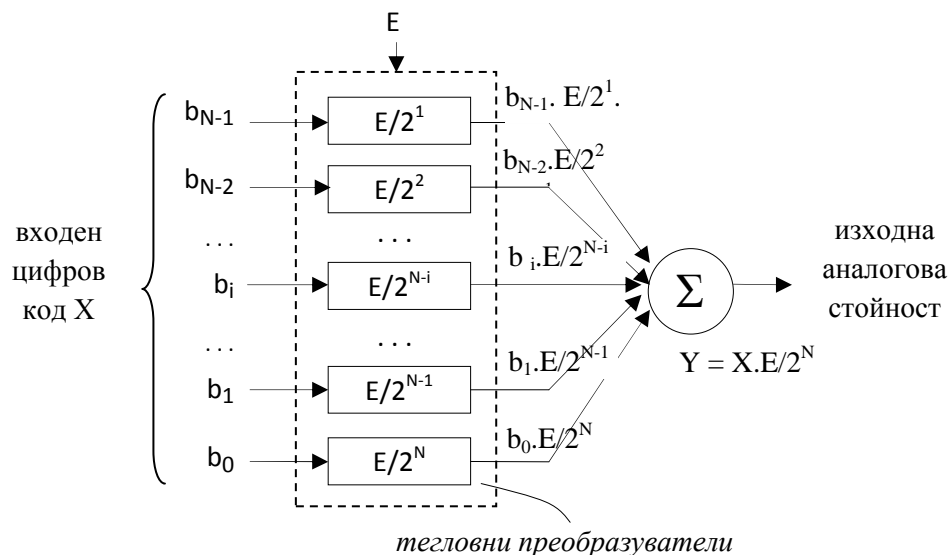
#### 1. ИЗВЕЖДАНЕ НА АНАЛОГОВИ СИГНАЛИ ЧРЕЗ ЕМК 68НС11

##### 1.1. Принцип на цифрово-аналоговото преобразуване

Числата от една позиционна бройна система (например, двоична - виж Упр. #1) могат да представят не само броя на отделни дискретни обекти, а и стойността на непрекъснати (аналогови) физически величини. За целта, на отделните разряди се съпоставят еталонни величини със съответно тегло и еталоните на разрядите, които имат стойност единица, се сумират. Това съответства на представянето на едно двоично число като сума от двоични еталони (двоично-десетично преобразуване) като най-малкото тегло има стойност единица, например:

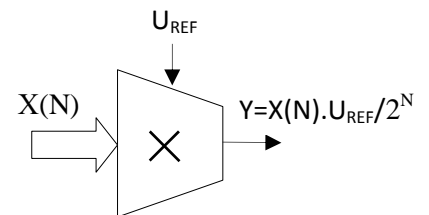
$$0101_2 = 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 + 4 + 0 + 1 = 5_{10}$$

Устройството, реализиращо тази операция (*цифрово-аналогов преобразувател* или съкратено, ЦАП) се състои от три части (фиг.1): източник на опорна величина E, тегловни преобразуватели E/2<sup>i</sup> и суматор Σ.



Фиг.1 Принцип на цифрово-аналоговото преобразуване

На практика един електрически ЦАП има два входа - *аналогов*, на който се подава опорното напрежение U<sub>REF</sub>, и *цифров*, на който се подава входния цифров код) и един *аналогов изход*, който генерира сигнал, пропорционален на произведението на двете входни величини (фиг.2). Затова цифрово-аналоговите преобразуватели с изведен вход за външно опорно напрежение се наричат още *умножаващи*. В повечето случаи това напрежение се поддържа неизменно по стойност от външен стабилизирани източник или се генерира вътрешно-схемно (но все пак присъства). Тогава изходното напрежение зависи само от подадения входен цифров код и този умножител действа като обикновен цифрово-аналогов преобразувател с един вход.

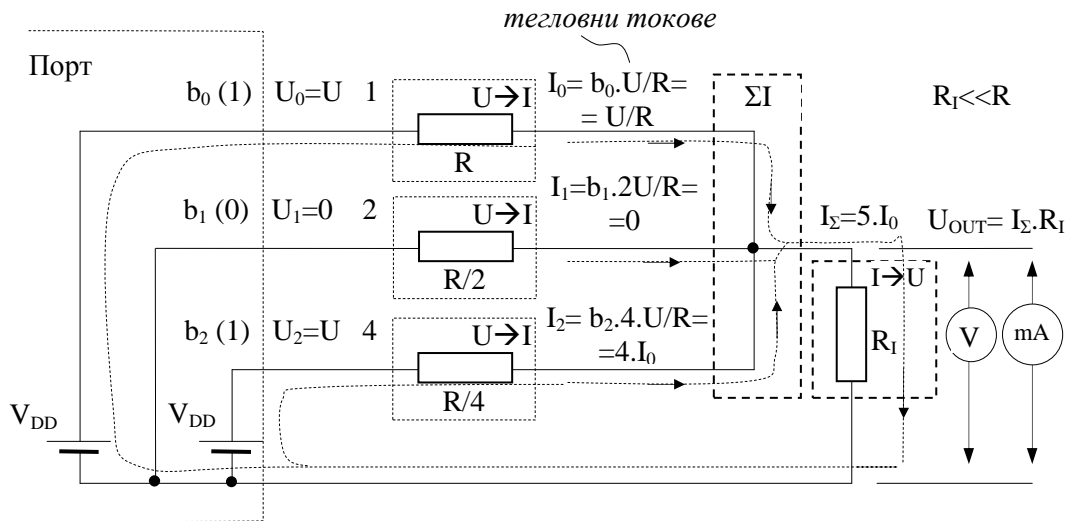


Фиг.2 Обобщена блок-схема на преобразуване на цифров код в аналогова изходна величина

##### 1.2. Цифрово-аналогови преобразуватели с тегловни резистори

В един елементарен електрически ЦАП (например, 3-битовия на фиг.3), изходите (b<sub>0</sub>, b<sub>1</sub>, b<sub>2</sub>) на управляващата цифрова схема (например, Порт В на ЕМК НС11), които

подават входния цифров код (в случая, 101), изпълняват ролята на управлявани опорни източници - ако даден бит е единица ( $b_0, b_2$ ) източникът е включен, а ако е нула ( $b_1$ ) – изключен. Тези опорни напрежения се преобразуват в тегловни токове ( $I_0, I_1$  и  $I_2$ ) посредством резистори (преобразуватели „напрежение в ток“) с двукратно намаляващо съпротивление ( $R, R/2$  и  $R/4$ ), които след това се сумират в изходния възел. Така на всеки входен бит съответства източник на ток със стойност, пропорционална на неговото тегло, а изходната величина е сума от токовете на всички разряди. Този изходен ток се преобразува в изходно напрежение  $U_{OUT}=5.I_0.R_1$  с помощта на резистор  $R_1$ , действащ като обратния преобразувател „ток-напрежение“. За намаляване на грешката, причинена от пада върху резистора  $R_1$ , той се избира с малко съпротивление, а за пълното ѝ премахване се включва инвертиращ операционен усилвател.



Фиг.3 3-битов ЦАП с тегловни резистори при зададени примерни входни стойности – графично представяне на тегловните токове

### 1.3. Включване на цифрово-аналогови преобразуватели към ЕМК 68HC11

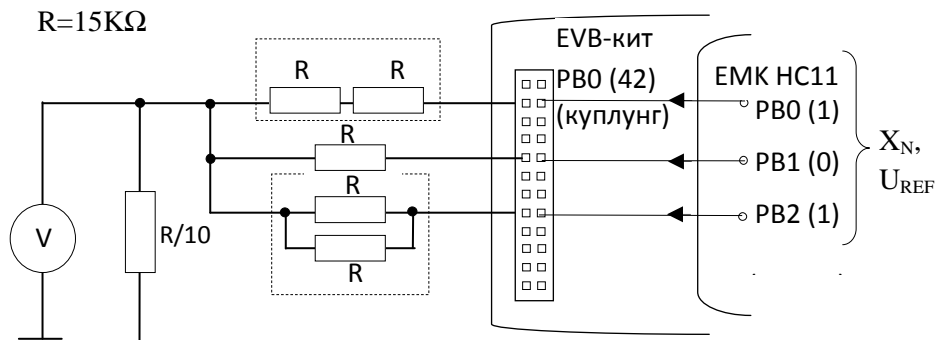
ЕМК 68HC11 няма вграден цифрово-аналогов преобразувател и затова, когато е необходимо, се включва допълнителен:

- специализиран ЦАП (с вътрешен регистър и вход за избор) към адресните и информационни шини на ЕМК, работещ в разширен режим, или
- обикновен ЦАП към паралелен порт на ЕМК (собствен или емулиран с HC24, както е направено в развойната система EVBplus2).

#### ✓ Регистър за данни PORTB (\$1004)

**Задача:** Монтирайте върху експерименталната платка (фиг.4) елементарния 3-битов ЦАП от фиг.3 като използвате резистори само с една стойност (виж. Приложение 1). Свържете входовете му към изходи PB0, PB1 и PB2 от Порт В. В изхода включете първо *токов консуматор* (стрелкови или цифров миламперметър, светодиоди, вход на биполярен транзистор и др.), а след това *напрежителен консуматор* (стрелкови или цифров волтметър, осцилоскоп, звуков индикатор, АЦП в следващата точка и др.) паралелно на нискоомен токов резистор  $R_1$  (напр.  $R_1=R/10$ ). Командвайте ЦАП като записвате различни числа в регистър PORTB (\$1004) в целия му диапазон и измервайте изходната величина (ток или напрежение) за всяка подадена стойност. Подадените числа можете да наблюдавате в двоична форма върху вътрешните светодиоди, свързани към изходите PB0, PB1 и PB2 от Порт В.

Какви стойности имат минималната и максималната изходна величина (ток или напрежение) и от какво се определят те? Защо не може да се достигне максималната стойност? Как може да се промени коефициента на предаване на преобразувателя? Какви са грешките на схемата, от какво зависят те и как могат да бъдат намалени или дори премахнати? Какъв е напрежителният пад върху резистора  $R_1$  - вреден или полезен?



Фиг.4 Схема на три-битов ЦАП с тегловни резистори с входен цифров код, подаван от Порт В

**Задача:** Напишете програма, която извежда линейно (стъпално) изменящо се напрежение в изхода на ЦАП от фиг.4, което се повтаря периодично с дадена честота. Наблюдавайте напрежението с помощта на осцилоскоп. Измерете амплитудата, нивото на стъпалото и честотата на сигнала. От какво зависят тези параметри на схемата? Променете ги като модифицирате програмата по подходящ начин. За да я спрете, натиснете бутона SW2 (RESET). За какво може да се използва тази постановка?

**Запазете схемата на ЦАП върху експерименталната платка за да я използвате при изследването на АЦП в следващата точка!**

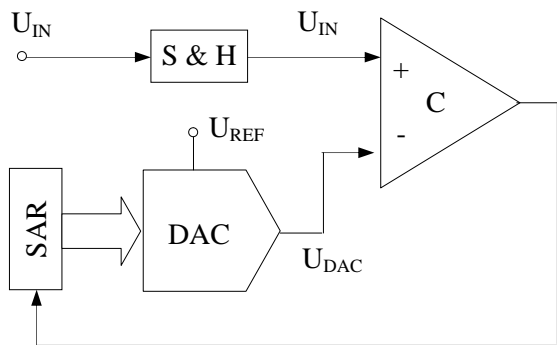
## 2. ВЪВЕЖДАНЕ НА АНАЛОГОВИ СИГНАЛИ ЧРЕЗ ЕМК 68НС11

### 2.1. Принцип на аналогово-цифрово преобразуване – метод на последователното приближение

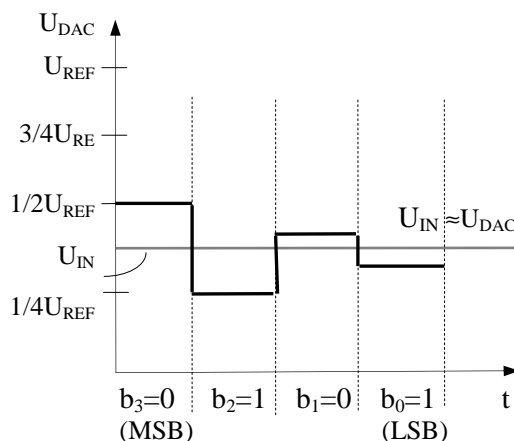
Непрекъснатите електрически величини (най-често, напрежение или ток) могат да се преобразуват в абстрактни двоични кодове (разгледани в Упр.1). За целта, на непрекъснатата величина се съпоставя еквивалентна сума от еталонни електрически величини със съответно тегло и в съответните разряди на цифровия код се записва единица, ако отговарящият му еталон участва в общата сума. Това съответства на десетично-двоичното преобразуване от Упр.1, осъществено чрез разлагане на десетичното число на сума от двоични еталони) като най-малкото тегло има стойност единица, например:

$$5_{10} = 0.2^3 + 1.2^2 + 0.2^1 + 1.2^0 = 0 + 4 + 0 + 1 = 0101_2$$

Устройството, реализиращо тази операция – *аналогово-цифров преобразувател* (АЦП) се състои от три части (на фиг.5А е показано неговата обобщена блокова схема): аналогова “памет” за входното напрежение - схема S&H “Sample and Hold” (следене-запомняне), цифрово-аналогов преобразувател DAC като източник на еталонно напрежение  $U_{DAC}$ , управляван от регистър за последователно приближение SAR, и сравняващо устройство (компаратор) С. Най-популярен метод за аналогово-цифровото преобразуване е *последователното приближение* (Successive Approximation, SA), известен още като *поразрядно кодиране* или *двоично търсене* (фиг.5Б – принцип на поразрядното кодиране за 4-битов АЦП). При него еталонните напрежения се включват/изключват последователно в посока от най-голямото ( $U_{REF}/2$ ) към най-малкото ( $U_{REF}/16$ ) така, че текущата им сума  $U_{DAC}$  да клони към входното напрежение  $U_{IN}$  и в края на процедурата почти да се изравни с него. Целта на процедурата е да се установи кои еталонни напрежения участват в нея - ако дадено еталонно напрежение присъства в сумата, в съответния разряд на регистъра SAR се записва единица, в противен случай - нула. Така след приключване на процедурата в регистъра се съдържа цифровият еквивалент на еталонното, съответно входното напрежение.



Фиг.5А АЦП с последователно приближение /обобщена схема/

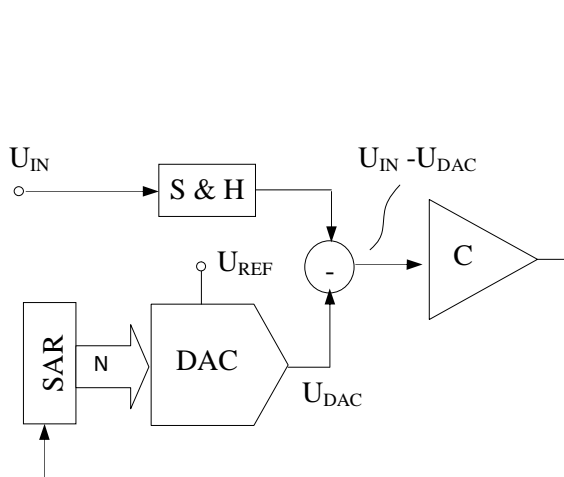


Фиг.5Б Принцип на последователното приближение: примерен 4-битов АЦП

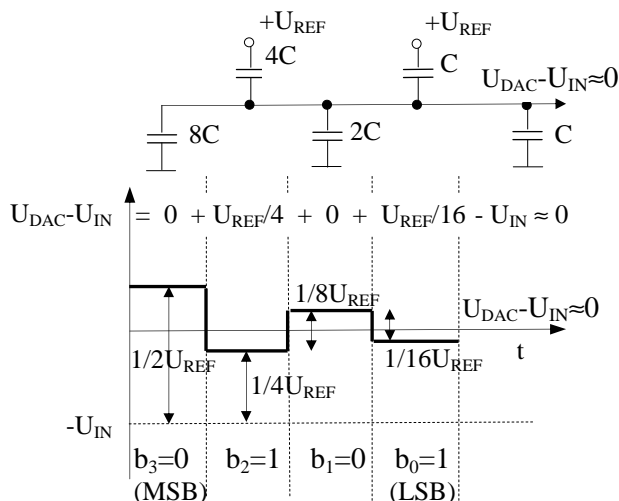
## 2.2. Аналогово-цифров преобразувател на ЕМК 68HC11

В ЕМК 68HC11 е използвана една разновидност на класическия метод на последователното приближение, при който с цел сравнение входната величина  $U_{IN}$  се изважда от еталонната величина  $U_{DAC}$  и в резултат на това, разликата им  $U_{DAC}-U_{IN}$  клони към нула. Тук източникът на еталонно напрежение (фиг.6А) е един кондензивен ЦАП (кондензивна „матрица”), съставен от двоично тегловни кондензатори с капацитет  $8C, 4C, 2C, C$  (и един изравнителен кондензатор  $C$ , най-вдясно), който едновременно изпълнява ролята и на схема „следене-запомняне” (фиг.6Б). Процедурата на последователното приближение протича в реда, показан на фиг.6Б.

В началото всички кондензатори са свързани паралелно и положителното входно напрежение се включва към тях, но „обърнато” с отрицателна полярност (за да може да бъде компенсирано впоследствие чрез положителното опорно напрежение). След това кондензаторите се откачат последователно (в посока от най-големия към най-малкия) с долния си извод от масата и се свързват към  $+U_{REF}$ .



Фиг.6А Обобщено представяне на принципа за АЦП, използван в ЕМК 68HC11



Фиг.6Б Определяне на изходния цифров код при 4-битов кондензивен SAR-АЦП

Когато е свързан към  $+U_{REF}$ , съответният кондензатор и еквивалентният кондензатор, образуван от всички останали паралелно свързани кондензатори, формират кондензивен делител със съответния двоично тегловен коефициент ( $0,5 \rightarrow 0,25 \rightarrow 0,125 \rightarrow 0,0625$ ). Двоично тегловното му напрежение ( $U_{REF}/2 \rightarrow U_{REF}/4$

→  $U_{REF}/8$  →  $U_{REF}/16$ ) се изважда от текущото напрежение върху „батерията” от кондензатори, свързани към маса (наслагват се две противоположни напрежения), и резултатното напрежение се проверява от компаратора. Ако напрежението е отрицателно ( $U_{REF} < U_{IN}$ ), кондензаторът се оставя свързан към +  $U_{REF}$ , а в съответния разряд се записва „1”, с което се отбелязва, че това „тегло” присъства в общата сума. Ако напрежението е положително ( $U_{REF} > U_{IN}$ ), кондензаторът се свързва обратно към маса, при което тегловното напрежение се прибавя обратно (възстановява се) към текущото напрежение; в съответния разряд се записва „0”, с което се отбелязва, че това „тегло” не участва в общата сума. Така, след приключване на процедурата, резултатното крайно напрежение  $U_{END}$  е почти равно на нула:

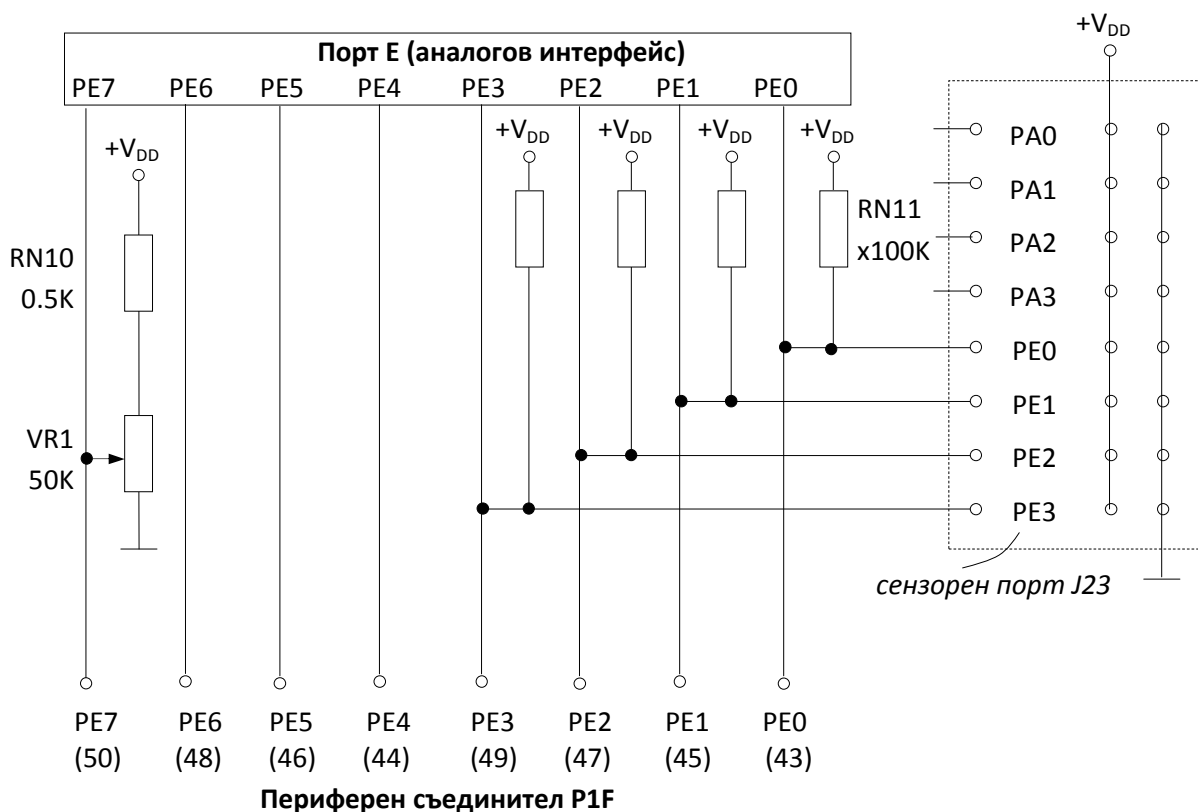
$$U_{END} = U_{DAC} - U_{IN} = 0 + U_{REF}/4 + 0 + U_{REF}/16 - U_{IN} \approx 0,$$

а в регистъра SAR се намира цифровият еквивалент на входното напрежение. Сумарният заряд в долните кондензатори е нула, а в горните е равен на началния заряд, обусловен от входното напрежение, приложено върху всичките паралелно свързани кондензатори в началото на процедурата. Зарядът сякаш се е преместил (преразпределил) от долните в горните кондензатори; затова тези преобразуватели са известни като АЦП с преразпределение на заряда (*charge redistribution*). Това може да бъде илюстрирано с аналогичната житейска задача *измерване на количество течност чрез изгребване с набор от еталонни съдове*.

### 2.4. Изпълнение на Порт Е върху системата за развитие EVBplus2

- ✓ Регистри – за състояние ADCTL (\$1030) и за данни ADR1÷ADR4 (\$1031÷1034)

Към входа на АЦП е включен аналогов мултиплексор, с помощта на който се превключват 8-те аналогови входа на Порт Е. Първите четири входа PE0÷PE3 са изведени (фиг.7) както към периферните съединители P1F и P1 (изводи 43, 45, 47 и 49), така и към сензорния порт (куплунг J23), разположен в долната част на кита (на него има и захранващите шини + $V_{DD}$  и маса). Те са свързани чрез високоомни (100K $\Omega$ ) pull-up резистори RN11 към захранващото напрежение + $V_{DD}$ , така че ако даден вход не е свързан към маса,



Фиг.7. Извеждане на изводи PE0-PE3 към куплунг PF1 и към сензорния порт, както и подаване на регулируемо напрежение през потенциометъра VR1 към извод PE7

напрежението му е равно на захранващото. Аналоговият интерфейс на ЕМК НС11 действа съгласно алгоритъма, показан в Приложение 3.

Резистивните сензори могат да бъдат включвани директно между тези входове и маса, без да е необходим допълнителен *pull-up* резистор. Следващите три входа PE4÷PE6 на Порт Е са изведени към изводи 44, 46 и 48 на куплунгите P1F и P1. Те нямат включени резистори така, че ако бъдат оставени „висящи“, ще възприемат случайни шумове. Последният аналогов вход PE7 на Порт Е е изведен към извод 50 на куплунгите P1F и P1, но към него е включен и плъзгачът на вътрешния 50KΩ потенциометър VR1, захранен с +V<sub>DD</sub> (разположен е в долната част на кита).

**Не свързвайте вход PE7 (извод 50 на P1F и P1) към +V<sub>DD</sub>, защото може да повредите потенциометъра VR1, ако плъзгачът му е завъртян в противоположното крайно положение (маса)! Можете ли да обясните защо?**

#### **2.4.1. Изследване на порт Е чрез измерване на напрежение от вътрешния потенциометър VR1 (едноканален режим на работа)**

##### ✓ Ръчен режим режим на изследване

**Задача:** Изследвайте действието на аналоговия Порт Е в режим на *еднократно измерване* на напрежението на вътрешния потенциометър VR1 чрез вход PE7. Включете контролен волтметър към извод 50 на периферния съединител P1F. Заредете директно (чрез командата ММ на монитора) необходимите стойности в съответните регистри и прочетете резултатите в следната последователност:

1. Включете захранването на АЦП чрез бит ADPU на регистър OPTION (\$1039).
2. Чрез запис в регистър ADCTL (\$1030) задайте режим *еднократно измерване* (бит SCAN), *единичен канал* (бит MULT) и *канал PE7* (битове CD÷CA) и стартирайте преобразуването (запомнете кода, защото ще ви се наложи да го въведете неколкократно).
3. Прочетете резултата от преобразуването от регистър ADR1 (\$1031).
4. Задайте нова стойност на входното напрежение като завъртите плъзгача на потенциометъра.
5. Повторете няколко пъти точки 2÷4 при различни стойности на напрежението.

Не трябва ли да проверявате бита за край на преобразуването CCF на регистър ADCTL след т.2? Какво се случва, ако забравите да изпълните т.2, а прочетете направо регистър ADR1?

**Задача:** Изследвайте действието на аналоговия Порт Е в режим на *непрекъснато измерване* на напрежението като направите съответните промени в горната последователност от действия.

**Съвет:** След като прочетете веднъж резултата от регистър ADR1 с командата ММ, натиснете и задръжте ENTER (функция REPEAT на клавиатурата) и изменяйте входното напрежение (въртете плъзгача на потенциометъра).

##### ✓ Програмен режим на изследване

Горната последователност от действия може да се изпълнява от проста програма като резултата от преобразуването се изпраща в Порт В за да се изобрази в двоичен вид от светодиодите, свързани към изходите му PB0÷PB7. В тази програма се използва функцията за непрекъснато повтарящо се преобразуване (бит SCAN=1). Проследете пълният алгоритъм на преобразуване в аналоговия интерфейс на НС11, даден в Приложение 2.

**Задача:** Заредете от папката D:\Students\Labs\Lab 6 (или въведете ръчно от листинга по-долу) текста на програмата за аналогово-цифрово преобразуване ADC1.asm в редактора на програмната среда AsmIDE. Разучете я като се опитате да разпознаете горните действия в нея. Асемблирайте програмата и заредете

обектния код в системата за развитие EVBplus2. Стартирайте програмата и изменяйте входното напрежение като въртите плъзгача на потенциометъра.

\* ADC1.asm --- Аналогово-цифрово преобразуване ---

\* Измерва непрекъснато напрежението на вход PE7 на Порт E (вътрешно свързан към

\* потенциометъра VR1) чрез режим SCAN на порта

\* Изпраща резултата в двоичен вид към порт B, индициран със светодиоди

\* Дефиниции на регистри

PORTB	EQU	\$1004	; Порт B (индициран със светодиоди)
ADCTL	EQU	\$1030	; Регистър на състоянието на АЦП
ADR1	EQU	\$1031	; Резултат от АЦ преобразуване
OPTION	EQU	\$1039	; Системен конфигуриращ регистър
	ORG	\$D000	

\* Инициализация на порта

	LDA	#%10000000	; Включва захранването на АЦП
	STAA	OPTION	; (ADPU = 1)
CONVERT	LDA	#%00100111	; Стартира непрекъснато повтарящо се преобразуване от
	STAA	ADCTL	; вход PE7 (SCAN=1, MULT=0, CC=CB=CA= 1)

\* Непрекъснато извлича и индицира резултата от преобразуването

RESULT	LDA	ADR1	; Чете резултата от преобразуването
	STAA	PORTB	; Изпраща го в порт B за индикация
	BRA	RESULT	; Отива да повтори четенето и изобразяването

**Въпроси:** Може ли цялото устройство (развойната система EVB, изпълняваща горната програма) да се разглежда като един възможен преобразувател на ъгъл на завъртане в двоичен код? На какво условие трябва да отговаря потенциометъра в това приложение? Защо е толкова трудно да зададете точно средно положение (плъзгача в средата, показание \$80)?

Програмата може да се модифицира така, че тя да стартира преобразуването, да изчаква приключването му и едва след това да прочита резултата.

**Задача:** Заместете съответната част от текста на горната програма с фрагмента по-долу (ако се затруднявате, заредете програмата ADC2.asm). Изследвайте действието ѝ. Можете ли да зададете точно средното положение и защо?

\* Извършва аналогово-цифрово преобразуване

CONVERT	LDA	#%00000111	; Подготвя задаване на еднократно измерване от вход PE7
	STAA	ADCTL	; (SCAN=0) и стартира преобразуването
WAIT	LDA	ADCTL	; Изчаква да свърши преобразуването
	BPL	WAIT	; (да се вдигне флагът CCF)

\* Извлича и индицира резултата от преобразуването

	LDA	ADR1	; Чете резултата от преобразуването
	STAA	PORTB	; Изпраща го в порт B за индикация
	BRA	CONVERT	; Отива да повтори операциите

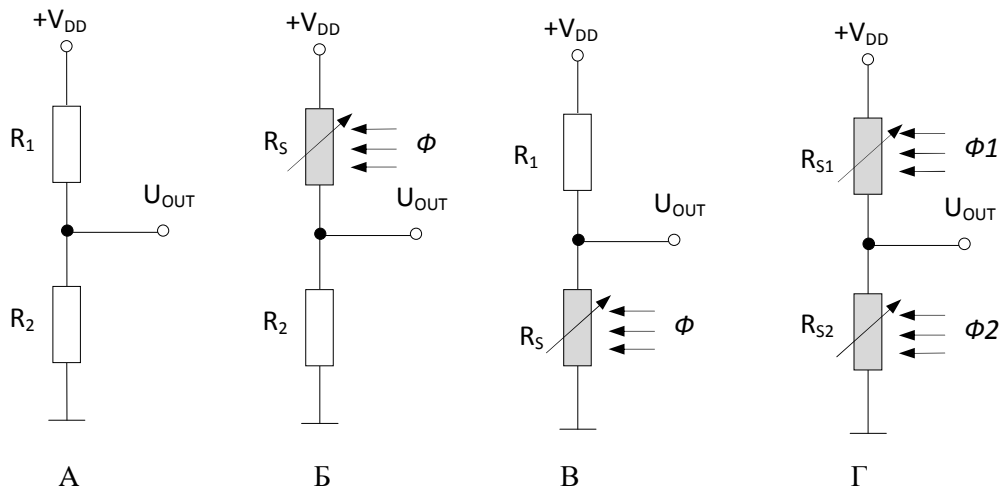
В програмите по-горе резултатът се индицираше в двоичен вид чрез светодиодите, свързани към Порт В. Ако към тях се добави програмата за динамична индикация от предишното упражнение, резултатът може да се индицира върху 7-сегментните индикатори в десетичен (BCD) вид.

**Задача:** Заредете текста на програмата ADC\_7S.asm (Приложение 4) в редактора на програмната среда AsmIDE и я разучете. По какво се различава тя от горните програми? Разделете я на познати части и ги изследвайте. Как става преобразуването от двоичен в BCD код? Как се определят мащабните коефициенти К? Защо се използват и четирите резултата от измерането на едно и също входно напрежение? Има ли смисъл да се отчитат и стотните? Разучете новите инструкции и директиви на асемблера. Асемблирайте програмата и заредете обектния код в системата EVBplus2. Стартирайте програмата и изменяйте входното напрежение като въртите плъзгача на потенциометъра.

#### 2.4.2. Изследване на порт Е чрез измерване на напрежение от външни сензори

##### ✓ Понятие за резистивни сензори

Това са двуполусни елементи (реостати, фоторезистори, терморезистори и др.), които имат свойството да променят електрическото си съпротивление адекватно на някаква неелектрическа величина – преместване, осветеност, температура и др. Тези промени на съпротивлението се преобразуват най-често в промени на напрежението в схемата на един делител на напрежение (фиг.8А), като сензорите се включват на мястото на резистора  $R_1$  (фиг.8Б),  $R_2$  (фиг.8В) или и на двата едновременно (фиг.8Г). Така се получават съответно права, обратна и диференциална зависимост на изходното напрежение от входните неелектрически величини.



Фиг.8. Включване на външни резистивни сензори (терморезистори, фоторезистори и др.) като част от делителя 8А в следните случаи: към захранващ източник (8Б), към маса (8В), едновременно към захранване и маса (8Г)

Могат да се използват различни сензори:

- вътрешния потенциометър (завъртане – напрежение);
- външен потенциометър (завъртане – напрежение);
- фоторезистор (светлина – съпротивление – напрежение);
- терморезистор (топлина – съпротивление – напрежение);
- активен термосензор (изисква усилвател);
- интегрираща RC верига (време – напрежение);



- изходът на ЦАП от предишната точка.

✓ **Едноканален режим на измерване**

**Задача:** Монтирайте последователно на експерименталната платка различни сензори (Приложение 2) като ги включвате между някой от входовете PE4÷PE6 и маса или ги свързвате в схемата на делител на напрежение, съгласно фиг.8. Въздействайте на сензорите по подходящ начин и измервайте изходното напрежение по някой от горните начини.

**Задача:** Напишете програма, под управлението на която се изследва автоматично една интегрираща RC верига.

*Съвет:* захранете веригата от изход на порт В.

✓ **Многоканален режим на измерване**

**Задача:** Монтирайте едновременно на експерименталната платка три различни сензора и ги включете към входовете PE4÷PE6. Модифицирайте горната програма така, че да сканира непрекъснато тези канали.

### 3. ЕДНОВРЕМЕННО ИЗВЕЖДАНЕ И ВЪВЕЖДАНЕ НА АНАЛОГОВИ СИГНАЛИ ЧРЕЗ 68HC11

**Задача:** Монтирайте отново върху експерименталната платка елементарния 3-битов ЦАП от фиг.4 като свържете входовете му към изходи PB0, PB1 и PB2 на Порт В, а изхода му – към някой от входовете PE4÷PE6 на Порт Е.

Задавайте входни числа в регистър PORTB (\$1004) и измервайте изходното напрежение на ЦАП с помощта на някой от горните програми.

---

#### II. ЗАДАЧИ ЗА ИЗПЪЛНЕНИЕ:

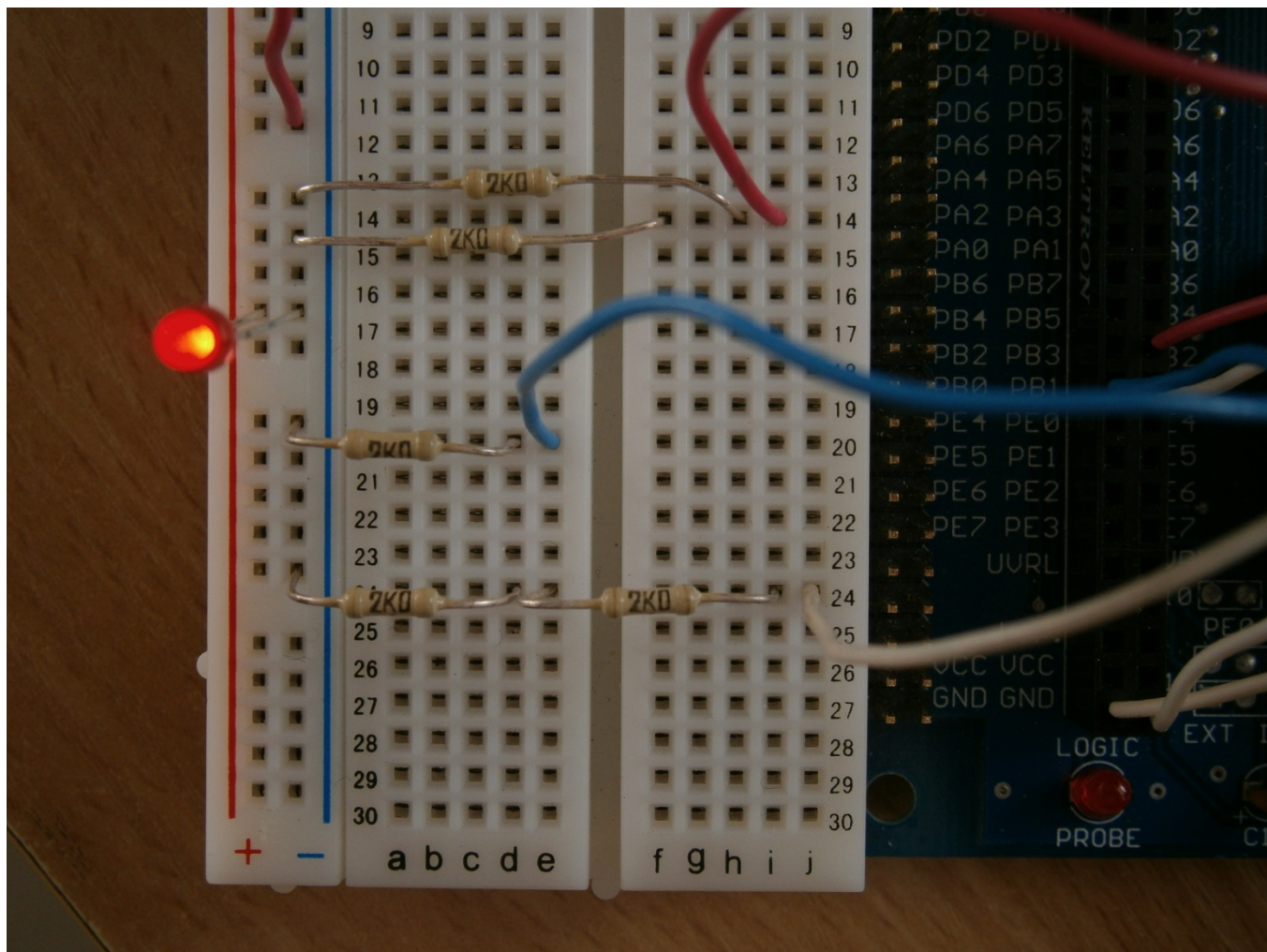
1. Запознайте се с аналоговия интерфейс (Порт Е на ЕМК HC11) и начина, по който е реализиран на системата за развитие EVBplus2. За какво са сложени pull-up резисторите на входове PE0÷PE3? Създават ли проблеми при измерване на аналогови сигнали?

2. Каква е ролята на опорното напрежение за действието на един ЦАП? От къде се получава и как се задава опорното напрежение в схемата на елементарния 3-битов ЦАП от фиг.4? Как може да го промените?

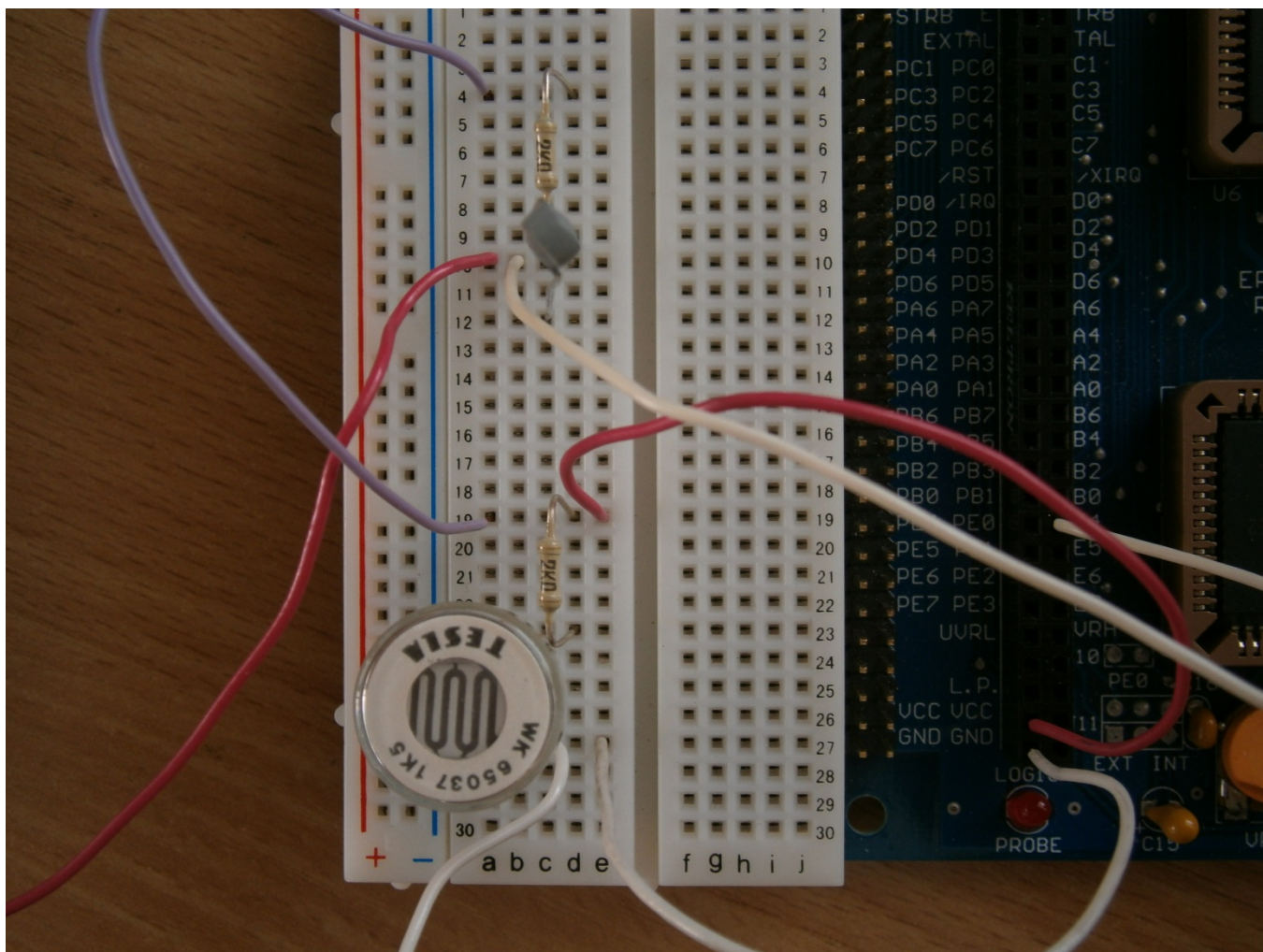
3. Каква е ролята на опорното напрежение за действието на едно АЦП? Проследете веригата, задаваща опорно напрежение на вграденото АЦП. Какви са възможностите за подаване на външно опорно напрежение и за какво може да се използва това? Може ли да използвате елементарния 3-битов ЦАП от фиг.4 за тази цел и какво ще постигнете с това?

4. С помощта на елементарния 3-битов ЦАП от фиг.4 и компаратор направете елементарно 3-битово АЦП, което измерва входно напрежение от потенциометър или сензор. Напишете програма на асемблер, която реализира метода на последователното приближение (фиг.5А) и изпраща резултата на Порт В (визуализира го със светодиодите).

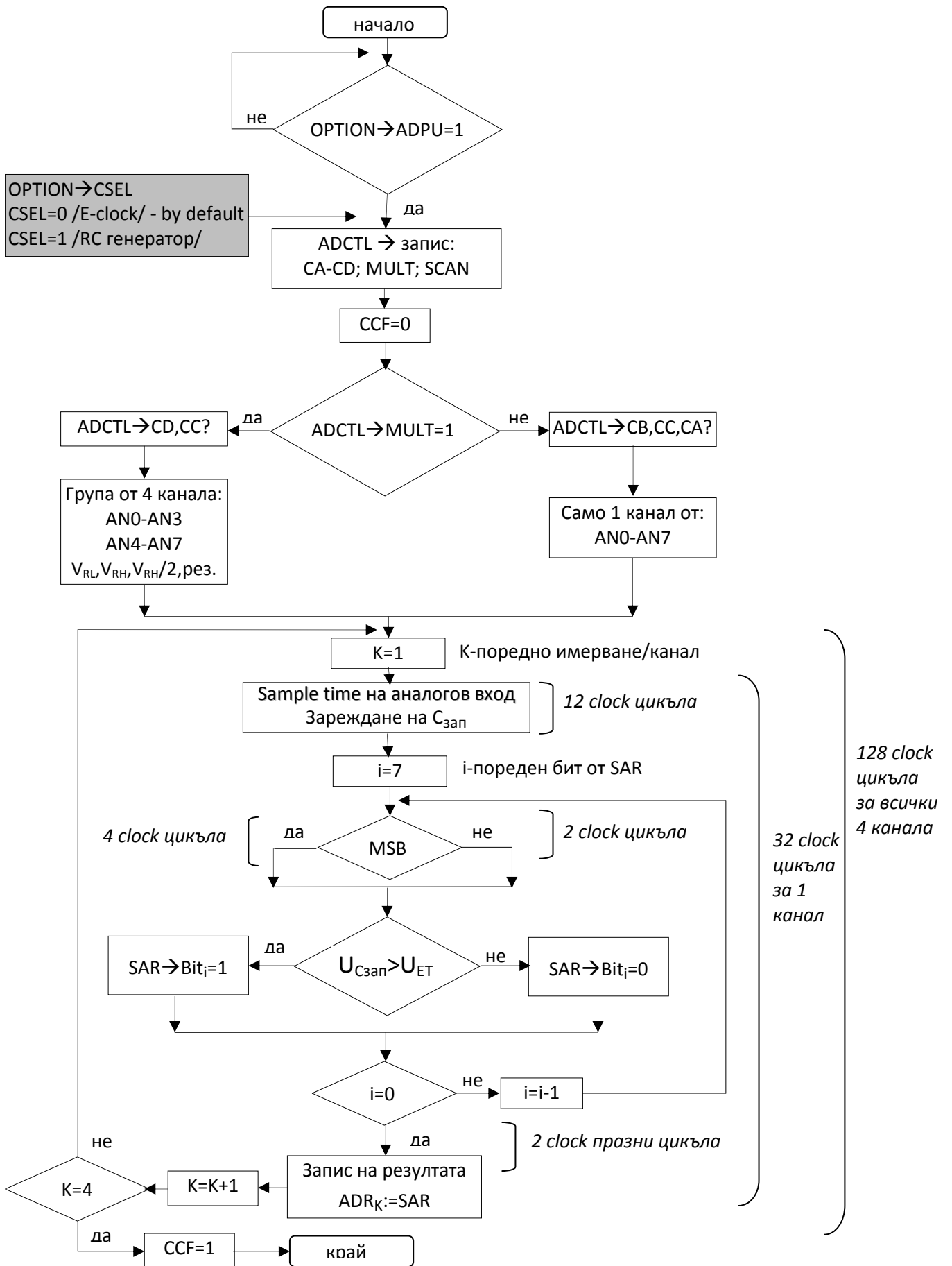
## Приложение 1

**3-битов ЦАП с тегловни резистори, монтиран върху експерименталната платка EVBplus2**

## Приложение 2

**Различни сензори (фоторезистор и термистор), монтирани върху експерименталната платка EVBplus2**

Алгоритъм на преобразуване на сигнали в аналоговия интерфейс на HC11



## Приложение 4

**Аналого-цифрово преобразуване в HC11 с изобразяване стойността на напрежението върху  
7-сегментните индикатори от кита EVBPlus2**

\* ADC\_7S.asm --- Аналогово-цифрово преобразуване в десетичен вид ---  
\*  
\* Измерва непрекъснато напрежението на вход E7 (вътрешно свързан към  
\* потенциометъра VR1) и индицира резултата върху 7-сегментните индикатори

\* Дефиниции на портове, обслужващи 7-сегментната индикация

PORTB	EQU \$1004	; Регистър данни порт B
PORTD	EQU \$1008	; Регистър данни порт D
DDRD	EQU \$1009	; Регистър посока порт D
ANODE	EQU PORTB	; Порт B управлява анодите на сегментите с "1" ; (PB0 - сегмент "a" ... PB7 - сегмент "g")
CATHODE	EQU PORTD	; Порт D управлява общите катода с активна "0" ; (PD5 - катод 3 ... PD2 - катод 0)

\* Дефиниции на регистрите на Порт E (АЦП)

OPTION	EQU \$1039	; Системен конфигуриращ регистър
ADCTL	EQU \$1030	; Регистър на състоянието на АЦП
ADR1	EQU \$1031	; Резултат 1 от АЦ преобразуване
ADR2	EQU \$1032	; Резултат 2 от АЦ преобразуване
ADR3	EQU \$1033	; Резултат 3 от АЦ преобразуване
ADR4	EQU \$1034	; Резултат 4 от АЦ преобразуване

ORG \$D100

\* Резервира памет за променливите

DELAY	FDB 10	; Време на светене на индикатора 10 mS
DIG_BCD	RMB 2	; Масив от 2-те цифри в BCD код
DIG_7S	RMB 2	; Масив от 2-те цифри в 7-сегментен код
DIG_POS	RMB 1	; Избран индикатор ("0" в съответната позиция)
ADC_SUM	RMB 2	; Сума от 4-те резултата

ORG \$D200

\* Описания на сегментите (цифра - 7-сегментен код)

SEGMTBL:		; . g f e d c b a
D0	FCB #\$3F	; "0" 0 0 1 1 1 1 1 1
D1	FCB #\$06	; "1" 0 0 0 0 0 1 1 0
D2	FCB #\$5B	; "2" 0 1 0 1 1 0 1 1
D3	FCB #\$4F	; "3" 0 1 0 0 1 1 1 1
D4	FCB #\$66	; "4" 0 0 0 0 0 1 1 0
D5	FCB #\$6D	; "5" 0 1 1 0 1 1 0 1
D6	FCB #\$7D	; "6" 0 1 1 1 1 1 0 1
D7	FCB #\$07	; "7" 0 0 0 0 0 1 1 1
D8	FCB #\$7F	; "8" 0 1 1 1 1 1 1 1
D9	FCB #\$6F	; "9" 0 1 1 0 1 1 1 1

\* ----- ГЛАВНА ПРОГРАМА -----

ORG \$D000

\* Инициализира Порт E

LDAA	##%10000000	; Включва захранването на АЦП (бит ADPU = 1)
STAA	OPTION	
LDAA	##%00000111	; Задава еднократно измерване от вход PE7 ; (SCAN = 0, MULT = 0, CC = CB = CA = 1)



STAA ADCTL ; и стартира преобразуването

\* Извиква поредица от подпрограми

BEGIN BSR ADC\_CONV ; Прави аналогово-цифрово преобразуване  
 BSR BCD\_CONV ; Преобразува резултата от в десетичен (BCD) вид  
 BSR IND\_7S  
 BRA BEGIN

\* ----- Подпрограми -----

ADC\_CONV: ; Извършва аналогово-цифрово преобразуване  
 WAIT\_CCF LDAA ADCTL ; Изчаква да свърши преобразуването  
 BPL WAIT\_CCF ; (флагът CCF да се вдигне)

\* Сумира резултатите (ADR1+ADR2+ADR3+ADR4) -> ADC\_SUM  
 CLRA  
 LDAB ADR1 ; Чете първия резултат от преобразуването  
 ADDB ADR2 ; Прибавя към него втория резултат  
 ADCA #0 ; Отчита евентуалния пренос  
 ADDB ADR3 ; Прибавя третия резултат  
 ADCA #0 ; Отчита преноса  
 ADDB ADR4 ; Прибавя четвъртия резултат  
 ADCA #0 ; Отчита преноса  
 STD ADC\_SUM ; Запомня сумата  
  
 LDAA #%00000111  
 STAA ADCTL ; Стартира отново преобразуването  
 RTS

BCD\_CONV: ; Преобразува резултата от АЦП в десетичен (BCD) вид

\* Определя цялата цифра D2 преди десетичната точка ("волтовете")

\* като дели учетворената сума в ADC\_SUM на мащабния коефициент K1 [ед/V]

LDD ADC\_SUM ; Взема учетворената сума  
 LDX #210 ; Зарежда мащабен коеф.  $K1 = 4.256/4,9 = 209$  [ед/V]  
 ; (опорното напрежение е 4,9 V)  
 IDIV ; Първата цифра ADC\_SUM/K1 -> инд. рег. X  
 STD ADC\_SUM ; Запазва остатъка  
 XGDX ; Слага първата цифра в мл. част на ACCD (ACCB)  
 STAB DIG\_BCD ; и я записва в DIG\_BCD

\* Определя цифрата D1 след десетичната точка ("десетите волтове")

LDD ADC\_SUM ; Взема остатъка  
 LDX #21 ; Зарежда мащ. коеф.  $K10 = 4.256/4,9/10 = 21$  [ед/100mV]  
 IDIV ; Първата (цялата) цифра [V] -> инд. рег. X  
 XGDX ; Слага втората цифра в мл. част на ACCD (ACCB)  
 STAB DIG\_BCD+1 ; и я записва в DIG\_BCD + 1  
 RTS

IND\_7S: ; Визуализира резултата в десетичен вид върху 7-сегментните индикатори

\* Инициализация

LDAA #%00011000 ; Зарежда "1" на съотв. места в регистъра за посока  
 STAA DDRD ; за да конфигурира PD4 и PD3 като изходи  
 STAA CATHODE ; Зарежда изключваща маска 00011000 в порт B, с която  
 ; гаси всички индикатори ("повдига" катодите до +5V)  
 LDAA #%11101111 ; Зарежда маска с "0" на съотв. място за да подготви  
 STAA DIG\_POS ; избирането на най-старшия индикатор (цифра 2)

\* Преобразува цифрите от двоичен в 7-сегментен код

```

LDY #DIG_BCD ; Началният адрес на масива от цифри в BCD код ->
; в индексния регистър Y
CONV_BCD LDAB 0,Y ; Поредната BCD цифра -> в акумулатор B
LDX #SEGMTBL ; Началният адрес на 7-сегментните дефиниции -> в
; индексния регистър X
ABX ; Прибавя отместване = стойността на BCD цифрата към
; съдържанието на инд. регистър X (така адресът на
; 7-сегм. код е вече в индексния регистър X)
LDAB 0,X ; Извлича 7-сегментния код и го помества в акум. B
STAB 2,Y ; 7-сегментният код -> на поредното място в DIG_7S
INY ; Подготвя следващия адрес на DIG_BCD
DECA ; Отброява още една цифра
BNE CONV_BCD ; Минава на следващата цифра, ако не са свършили

```

\* Извежда последователно 4-те цифри в 7-сегментен код върху индикаторите

```

LDX #DIG_7S ; Адресът на първата цифра в 7-сегм. код -> в X
; (да сочи адреса на първия елемент от DIG_7S)
BSET 0,X %10000000 ; Слага "." (установява бит 7 на DIG_7S в "1")
IND1 LDAA #%00111100 ; Зарежда изключваща маска ("1" на всички катода), с
STAA CATHODE ; която гаси всички индикатори ("повдига" потенциала
; на катодите им до +5V)
LDAA 0,X ; Извлича поредния 7-сегментен код от масива DIG_7S
STAA ANODE ; Подготвя включването на съответните сегменти
; като подава +5V (през резисторите) към анодите
INX ; Подготвя избирането на следващия 7-сегментен код
LDAA DIG_POS ; Чете номера на текущата цифра (катод)
STAA CATHODE ; Избира текущия катод (подава 0V), протича ток през
; сегментите и цифрата светва

```

\* Задържа светенето на индикатора за време, зададено в клетките DELAY [ms]

```
LDY DELAY
```

\* Външен цикъл - задържа за време 1ms

```
DLY11 LDAB #220
```

\* Вътрешен цикъл - задържа за време 4,5µS

```

DLY21 NOP ; 1µS
NOP ; 1µS
DECB ; 1µS
BNE DLY21 ; 1,5µS
DEY ; Отброява още една милисекунда
BNE DLY11 ; Ако не е изтекло зададеното време, се връща отново

```

```
LDAA DIG_POS
```

```

SEC ; Осигурява запълването на маската с "1" отляво
RORA ; Премества маската (нулата в нея) една стъпка надясно
BITA #$04 ; Проверява дали маската (нулата) е стигнала крайно
; положение (дали са обработени всичките 2 цифри)
BEQ IND2 ; Ако са обработени, излиза от подпрограмата
STAA DIG_POS ; Не са, установява маската в съответната позиция
BRA IND1 ; Отива да обработва следващата цифра

```

\* Всички цифри са изобразени, установява в изходна позиция и излиза

```

IND2 LDAA #%11101111 ; Подготвя установяването на маската в изх. позиция
LDX #DIG_7S ; Зарежда адреса на първата цифра в 7-сегм. код в X
RTS

```