

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 1</i>

ЗАТВЕРДЖЕНО

Науково-методичною радою
Державного університету
«Житомирська політехніка»

протокол від 09 грудня 2021 р.
№ 7

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

для проведення практичних занять
з навчальної дисципліни

«Цифрові та телевізійні мережеві технології»

для здобувачів вищої освіти освітнього ступеня «магістр»
спеціальності 172 «Телекомунікації та радіотехніка»
освітньо-професійна програма «Телекомунікації та радіотехніка»
факультет інформаційно-комп'ютерних технологій
кафедра біомедичної інженерії та телекомунікацій

Рекомендовано на засіданні
кафедри біомедичної інженерії та
телекомунікацій
26 серпня 2021 р., протокол №10

Розробник: к.т.н., доцент кафедри біомедичної інженерії та
телекомунікацій ЦИПОРЕНКО Валентин

Житомир
2021

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 2</i>

Зміст

Практична робота №1. Дослідження принципів дискретизації неперервних радіосигналів	3
Практична робота №2. Дослідження основних властивостей аналого-цифрового перетворення радіосигналів	15
Практична робота №3. Дослідження методів формування звукових сигналів з використанням МК AVR	22
Практична робота №4. Дослідження методів опізнавання сигналів з використанням мікроконтролерів AVR серії ATtiny	47
Практична робота №5. Дослідження функціональних можливостей таймерів	85
Практична робота №6. Дослідження функціональних можливостей пам'яті МК	95

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 3

Практична робота №1

Дослідження принципів дискретизації неперервних радіосигналів

Мета роботи: експериментальне дослідження основних принципів дискретизації неперервних радіосигналів та основних властивостей дискретних послідовностей.

1. Підготовка до виконання роботи.

- 1.1. Опрацювати необхідний теоретичний матеріал в рекомендованому переліку літератури.
- 1.2. Опрацювати методичні вказівки до проведення лабораторної роботи, уявити мету роботи та принцип її виконання.
- 1.3. Вивчити технічні описи приладів, що входять до складу лабораторного робочого місця, а також спеціалізованого пакету прикладних програм ПЕОМ.
- 1.4. Виконати необхідні попередні розрахунки.
- 1.5. Відповісти на питання самотестування, готовності до виконання роботи.
- 1.6. Для кожного пункту третього розділу проробити та запропонувати свій варіант методики досліджень в загальному вигляді із обґрунтуванням використання конкретної моделюючої програми із пакету прикладних програм. Варіант методики досліджень узгодити із викладачем.
- 1.7. Вибрати тип і структуру моделюючих сигналів та розрахувати конкретні їх параметри і діапазони їх можливих значень. Підготувати програму модель із урахуванням проведених розрахунків та узгодити її з викладачем.
- 1.8. Проаналізувати можливість розширення досліджень стосовно особливостей процесів дискретизації радіосигналів. Запропонувати варіанти додаткових досліджень та

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 4</i>

відповідні методики їх проведення в доповнення програми розділу 3.

2. Короткі теоретичні відомості

Для обробки радіосигналів за допомогою цифрових обчислювальних пристроїв необхідно їх попередньо перетворити в цифрову форму. Таке перетворення зазвичай виконується в два етапи. На першому етапі реалізується дискретизація в часі, а на другому – квантування по рівню.

В загальному випадку дискретизація неперервної функції часу складається у вимірюванні її значень в моменти часу, які рознесені один від одного на інтервал T_D , який зазвичай називають періодом дискретизації. Період дискретизації вибирають або постійним $t_k = k \cdot T_D$, або змінним. У другому варіанті змінний період дискретизації $T_D(t)$ є випадковою або детермінованою функцією часу.

Пристрій часової дискретизації можливо розглядати як перемикач, що замикається на час τ із періодом T_D . В результаті на виході перемикача із вхідного неперервного сигналу $S(t)$ формується послідовність з миттєвих значень (виборок) $S(t_k)$, що відповідають певним дискретним відлікам часової шкали. Часові діаграми, що пояснюють принцип дискретизації наведені на рис. 1.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 5

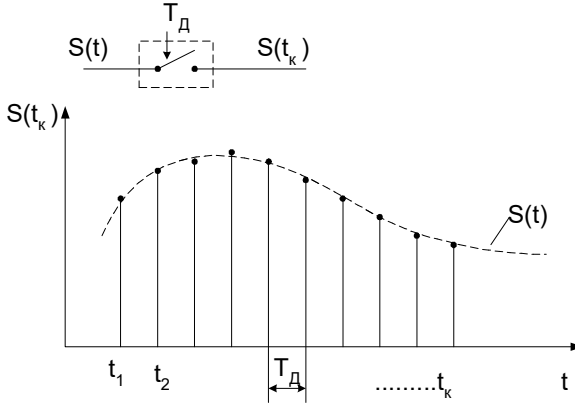


Рис. 1. Принцип дискретизації

При ідеальній дискретизації час спостереження (замикання перемикача) τ сигналу в момент часу t_k є нескінченно малий, тобто $\tau \rightarrow 0$. В цьому випадку дискретизований сигнал $S(t_k)$ можливо представити як послідовність δ -імпульсів (дельта-імпульсів, імпульсів Дірака), вага яких рівна миттєвим значенням сигналу $S(t)$ в дискретні моменти часу t_k :

$$S(t_k) = \sum_{k=0}^{\infty} S(t_k) \cdot \delta(t - t_k) \quad (1)$$

Представлення неперервного сигналу $S(t)$ у вигляді послідовності $S(t_k)$ можливо тільки із певними обмеженнями. Основним таким обмеженням є вимога обмеженості спектра сигналу $S(t)$, що дискретизується. В цьому випадку згідно теореми відліків (теорема Шеннона, теорема Котельникова), неперервний сигнал $S(t)$ із обмеженим спектром повністю визначається зліченою множеною дискретних виборок (значень), що рознесені один відносно одного у часі на величину $T_D \leq \frac{1}{2} f_{\max}$ (f_{\max} - гранична частота спектра

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 6

вихідного сигналу $S(t)$). При цьому згідно теореми відліків (теореми Котельникова-Шеннона) сигнал $S(t)$, що обмежений по спектру найвищою частотою f_{\max} , можливо представити рядом:

$$S(t_k) = \sum_{k=0}^{\infty} S\left(\frac{K}{f_{\max}}\right) \cdot \frac{\sin 2\pi f_{\max}(t - K/2f_{\max})}{2\pi f_{\max}(t - K/2f_{\max})} = \sum_{n=-\infty}^{\infty} S(KT_D) \cdot \varphi_K(t) \quad (2)$$

де $T_D = \frac{1}{2f_{\max}}$ - період дискретизації;

$S(K/2f_{\max}) = S(KT_D)$ - вибірки сигналу $S(t)$ в момент $t_k = K \cdot T_D$.

На практиці в радіоелектронних системах дискретизації підлягають сигнали, що обмежені в часі і мають тривалість T_c . В цьому випадку визначають найвищу частоту спектра f_{\max} , при якій похибка дискретизації за рахунок обмеження спектру сигналу була не більше допустимого значення. В результаті сигнал $S(t)$ повністю задається обмеженою сукупністю відліків $S(t_k)$ кількість яких дорівнює N :

Обмежений у часі сигнал $S(t)$ представляється рядом:

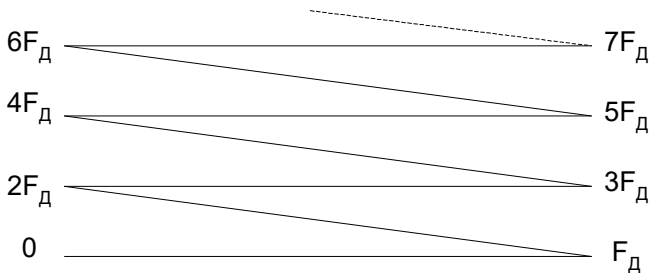
$$N = \frac{T_c}{T_D} = T_c \cdot 2f_{\max}. \quad (3)$$

$$S(t_k) = \sum_{K=0}^{2f_{\max} \cdot T_c} S(K \cdot T_D) \cdot \frac{\sin 2\pi f_{\max}(t - K \cdot T_D)}{2\pi f_{\max}(t - K \cdot T_D)}. \quad (4)$$

При дискретизації періодичного сигналу $S(t)$ може виникнути явище підміни частот, при якому вибірки сигналів $S_1(t)$ і $S_2(t)$, які відрізняються тільки частотою (періодом) повторення, можуть бути однаковими. Явище підміни частот доцільно досліджувати за

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 7

допомогою діаграми підмін, яка утворюється шляхом складання сегментами вісі частот із періодом $F_D = \frac{1}{2 \cdot T_D}$, рис.2.



На рис.2 показані перші сім складок діаграми підмін. Можливість підмини частот необхідно враховувати при виборі частоти дискретизації F_D .

В частотній області визначення ефект підмини частот пояснюється явищем цифрового гетеродинування. Математично цифрове гетеродинування спектру вхідного сигналу $S(t)$ при його дискретизації із частотою F_D можна бачити із виразу спектра дискретизованого сигналу $S(t_k)$:

$$S_D(\omega) = \frac{1}{T_D} \cdot \sum_{n=-\infty}^{+\infty} S_f \left(\omega - n \frac{2\pi}{T_D} \right) \quad (5)$$

де $S_f(\omega)$ - спектр початкового неперервного сигналу $S(t)$.

Із рівняння (5) витікає, що спектр дискретизованого сигналу $S(t_k)$ представляє собою періодичне повторення спектрів вхідного сигналу $S(t)$ із періодом, що дорівнює частоті дискретизації.

Якщо неперервний сигнал $S(t)$ має обмежений спектр, що дорівнює нулю за межами $\{f_s, f_n\}$ додатних частот, то його можливо

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 8

дискретизувати із частотою $F_{Д} \geq 2f_{\text{г}}$ або із частотою $F_{Дс}$, що задовольняє нерівність:

$$\frac{(f_{\text{сеп}} / \Delta f_c) + 1}{K + 1} < \frac{F_{Дс}}{2\Delta f_c} < \frac{(f_{\text{сеп}} / B) - 1}{K} \quad (6)$$

де $f_{\text{сеп}} = \frac{f_n + f_{\text{г}}}{2}$ - середня частота спектру сигналу;

$\Delta f_c = f_{\text{г}} - f_n$ - ширина спектру сигналу;

K - певне ціле число.

Явище, при якому використовують ефект цифрового гетеродинування і частоту дискретизації $F_{Дс}$, що менша частоти $F_{Д}$, називають субдискретизацією. Використання субдискретизації дозволяє зменшити число дискретних значень сигналу, що обробляються, а також спростити апаратно пристрої цифрової обробки сигналів.

При цифровій обробці радіосигналів об'єктом дискретизації у часі є випадковий процес $S_{\Pi}(t)$ на виході аналогової частини приймача. Основною особливістю цього процесу $S_{\Pi}(t)$ є його вузькосмуговість ($\Delta f_c / f_{\text{сеп}} \ll 1$). В цьому випадку для представлення сигналу $S_I(t)$ доцільно використовувати метод огиначаючих, у відповідності до якого вузько смуговий радіосигнал представляють у вигляді:

$$S_I(t) = U(t) \cos[2\pi f_0 t + \Psi(t)] \quad (7)$$

де $U(t)$ - низькочастотний сигнал (огиначаюча);

$\Psi(t)$ - закон фазової модуляції;

f_0 - несуча частота, $f_0 = f_{\text{сеп}}$.

В результаті часової дискретизації підлягають дві складові $U(t)$ і $\Psi(t)$, які змінюються у часі повільно із частотами:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 9

$$F_{Д1} \geq f_{U_{\max}}, F_{Д2} \geq f_{\Psi_{\max}} \quad (8)$$

де $f_{U_{\max}}, f_{\Psi_{\max}}$ - найвищі частоти в спектрах огинаючої $U(t)$ та миттєвої фази $\Psi(t)$, або двох квадратурних складових:

$$U_c(t) = U(t) \cos \Psi(t) \text{ і } U_s(t) = U(t) \sin \Psi(t)$$

Важливо відмітити, що частоти $f_{U_{\max}}$ і $f_{\Psi_{\max}}$ можуть бути різними, тобто $F_{Д1} \neq F_{Д2}$.

При обробці сигналів на проміжній частоті на виході радіоприймача доцільно враховувати параметри його амплітудно-частотної характеристики, що дозволяє вибрати оптимальне значення частоти дискретизації $F_{Д}$.

Вибір значення частоти дискретизації суттєво залежить від типу подальшої обробки дискретної послідовності $S(t_k)$. Основними варіантами алгоритмів обробки дискретизованого сигналу є:

- 1) вимірювання часових параметрів (час прийому, затримка надходження, тривалість);
- 2) вимірювання параметрів спектра або частотних параметрів (ширина спектра, несуча частота, тощо);
- 3) обробка сигналу, що враховує форму сигналу (задачі розпізнавання, радіозв'язок, радіомовлення тощо);
- 4) вимірювання миттєвих значень сигналу з метою автоконтролю, регулювання та сигналізації;
- 5) виявлення цілей, та прогнозування параметрів траєкторій цілей в радіолокації.

В кожному із вказаних випадків значення частоти дискретизації або періоду дискретизації вибирають із врахуванням певних критеріїв та параметрів радіосигналу.

Розглянемо основні варіанти обробки радіосигналів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 10

- 1) Якщо сигнал стаціонарний і має обмежений спектр, то період дискретизації вибирають за умов теореми відліків:

$$T_D \leq 1/2 f_{\max} .$$
- 2) Якщо сигнал нестаціонарний випадковий, із обмеженою тривалістю T_c , тоді період дискретизації не повинен перевищувати інтервалу кореляції $\tau_{кор}$ цього сигналу

$$T_D \leq \tau_{кор}$$
- 3) Якщо обробка сигналу направлена на отримання інформації, що передається, тоді період дискретизації вибирають із умови забезпечення допустимих втрат інформації $\sigma_{x\min}^2$ (дисперсії похибки) $\sigma_{x\min}^2 \leq (6\alpha - 1)/12\alpha^2$, де $\alpha = 1/\tau_{кор} \cdot T_D$
- 4) При обробці радіолокаційних сигналів в кожному періоді зондування не ставиться задача відновлення вихідного процесу після його дискретизації. Основною є вимога не пропустити корисний сигнал. В цьому випадку період дискретизації повинен бути не більше ефективної тривалості відбитого радіосигналу: $T_D \leq \tau_c$
- 5) При забезпеченні роздільної здатності РЛС по координаті r , період дискретизації повинен бути не більше тривалості зондуючого сигналу $T_D \leq \tau_c$
- 6) При забезпеченні необхідної точності дальності до цілі імпульсної РЛС, період дискретизації дорівнює $T_D \leq \sqrt{6D_z}$, де D_z - дисперсія допустимої помилки вимірювання дальності
- 7) При забезпеченні необхідної якості функціонування слідкуючих радіосистем період дискретизації дорівнює:

$$T_D \leq \varepsilon / V_{c\max} ,$$

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 11

- де ε - допустима помилка слідкування, $V_{\text{сmax}}$ - максимальна швидкість зміни радіосигналу
- 8) При вимірюванні параметрів спектра радіосигналу період дискретизації T_d визначають з урахуванням ефекту цифрового гетеродинування, та параметрів передаточної частотної характеристики лінійного тракту радіоприймача [].
- 9) При аналізі форми сигналу у часі період дискретизації вибирають з урахуванням допустимої похибки дискретизації [], що визначається обмеженістю часу аналізу сигналу та не ідеальністю відновлюючого фільтра. За критерієм період дискретизації вибирають в 2-5 разів більшим, ніж за теоремою відліків.

3. Порядок виконання роботи

3.1. Дослідити залежність похибки вимірювання амплітуди, часу надходження, тривалості та періоду слідування прямокутного відео імпульсу від частоти дискретизації стосовно цифрової радіосистеми вимірювання дальності до цілі. Для виконання пункту 3.1 використовувати програму *Numeri*, пункт 5-1(данние)-1-4(генератор стандартних сигналів). Результати дискретизації представляти в табличній формі. За таблицею визначити задані параметри для трьох форм сигналу. При обробці прямокутного сигналу інформаційним параметром використовувати стрибкоподібну зміну амплітуди. При трикутному та гармонічному сигналі – зміну знаку миттєвих значень сигналу, або краще знаку їх похідної.

3.2. Виконати дослідження п. 3.1 для умов наявності адитивного нормального шуму при відношенні с/ш=0, 10, 20, 30дБ. ($X_{\text{дБ}}=20\lg(A_{\text{с}}/A_{\text{ш}})$).

3.3. Дослідити в часовій області визначення ефект підміни частот для періодичних сигналів форми. За результатами досліджень

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 12

побудувати діаграму підмін. Для цього використовувати інтерпретатор формул (5-1-1-1), вибираючи F_d , значення якої менше F_c в $K=2,3,4$ разів.

3.4. Дослідити в частотній та часовій областях визначення ефект цифрового гетеродинування при дискретизації сигналів. Дослідити режим прямого та інверсного гетеродинування. Зняти залежність кратності гетеродинування K_{zem} від співвідношення частоти гармонічного та полігармонічного (сума 3-х синусоїд в інтерпретаторі формул) сигналу, та частоти його дискретизації. В частотній області визначення використовувати результати ДПФ із вікном Блекмана. Для цього використовувати інтерпретатор формул (5-1-1-1).

Щоб визначити наявність прямого гетеродинування потрібно спочатку визначити спектр сигналу при $F_d > 2F_c$ і зафіксувати таблицю відліків його прямої (на початку таблиці) та інверсної частини, що є дзеркальним відображенням прямої з інверсними фазами. Потім вибирати F_d , значення якої менше F_c в $K=2,3,4$ разів. При прямому гетеродинуванні пряма частина спектра не зміниться. При інверсному гетеродинуванні пряма та інверсна частини спектра поміняються місцями.

3.5. Дослідити в частотній області залежність значення частоти дискретизації радіосигналу на проміжній частоті в залежності від параметрів передаточної АЧХ приймального радіо тракту. Експериментально визначити залежність мінімально допустимого значення частоти дискретизації від параметрів передаточної АЧХ приймального тракту (коэф. Прямокутності АЧХ ($A_c(-3дБ)/A_c(-30дБ)$), селективності-рівня подавлення сусіднього каналу). Параметри АЧХ тракту задавати через параметри сигналів на його виході за умови рівномірного вхідного амплітудного спектра (аналізувати спектр суми 4-х синусоїд: $f_{c1}=5, A_1=A_{max}/30$; $f_{c2}=10, A_2=A_{max}$; $f_{c3}=20, A_3=A_{max}$; $f_{c4}=25, A_4=A_{max}/30$).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 13</i>

4. Обробка результатів досліджень

- 4.1. За результатами кожного пункту програми виконання роботи (розділ 3) побудувати діаграми або графіки залежностей похибок вимірювань або параметрів радіосигналу від періоду дискретизації та параметрів сигналу або тракту радіосистеми.
- 4.2. Виконати аналіз отриманих результатів та визначити оптимальні значення параметрів процедури дискретизації для умов досліджень.
- 4.3. Використовуючи відомі методи розрахувати теоретичні значення параметрів сигналів та процедури дискретизації і порівняти їх з експериментальними даними.
- 4.4. За результатами досліджень зробити узагальнюючі висновки та розробити відповідні рекомендації.
- 4.5. Результати досліджень оформити у вигляді звіту, що підлягає індивідуальному захисту у викладача.

Контрольні питання самоконтролю

1. Яка сутність процесу дискретизації і його математичні моделі?
2. Як визначити період дискретизації згідно теореми відліків?
3. В чому полягає процес цифрового гетеродинування при дискретизації сигналів?
4. В чому полягає ефект підміни частоти при дискретизації сигналів?
5. Як вибирається період дискретизації вузько смугових радіосигналів?
6. Як вибирається період дискретизації випадкових радіосигналів?
7. Які фактори впливають на вибір періоду дискретизації радіосигналів в радіотехнічних системах?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 14</i>

8. Які параметри радіоприймального тракту впливають на період дискретизації його вихідного радіосигналу?
9. В чому сутність прямого та інверсного цифрового гетеродинування при дискретизації радіосигналів?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 15

Практична робота №2

Дослідження основних властивостей аналого-цифрового перетворення радіосигналів

Мета лабораторної роботи: експериментальне дослідження основних властивостей аналого-цифрового перетворення (АЦ-перетворення) радіосигналів.

1. Теоретичні відомості

Аналого-цифрове і цифро-аналогове перетворення.

Сигнали можуть бути: 1) аналогові; 2) дискретні; 3) квантовані; 4) цифрові.

Аналогові сигнали: задаються по вісі часу на незліченій множині точок і на вісі рівня – незліченою множиною точок.

Дискретні сигнали: задані на зліченій (дискретній) множині точок на вісі часу і на незліченій множині значень по рівню.

Квантовані сигнали: задані на зліченій (дискретній) множині точок по рівню і на незліченій множині значень часу.

Цифрові сигнали: задані на злічених множинах відліків по рівню і по часу.

Аналоговий сигнал можна перетворити в цифровий за допомогою процедури аналого-цифрового перетворення. В свою чергу аналого-цифрове перетворення можна уявити як послідовність двох операцій: 1) дискретизації по часу; 2) квантування по рівню. Зазвичай результати перетворення кодуються з використанням прямого, зворотного або доповняльного кодів.

Цифровий сигнал можна перетворити в аналоговий за допомогою процедури цифро-аналогового перетворення. Використовується також термін «відновлення сигналу».

При аналого-цифровому перетворенні необхідно забезпечити однозначну відповідність попереднього аналогового сигналу і його цифрового еквіваленту. Ця вимога виконується, якщо період дискретизації буде вибраний на основі теореми Котельникова (Найквіста) або теореми відліків.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 16</i>

У відповідності з теоремою відліків неперервний сигнал $S(t)$ в спектрі якого не містяться спектральні складові з частотою вище f_{\max} , повністю описується вибірковими значеннями $S(n)$, відрахованими через інтервал $T_d \leq 1 / 2f_{\max}$. Аналітично це виражається у вигляді ряду Котельникова.

2. Підготовка до виконання роботи:

- 1.9. Опрацювати необхідний теоретичний матеріал в рекомендованому переліку літератури.
- 1.10. Опрацювати методичні вказівки до проведення лабораторної роботи, уяснити мету роботи та принцип її виконання.
- 1.11. Вивчити технічні описи приладів, що входять до складу лабораторного робочого місця, а також спеціалізованого пакету прикладних програм ПЕОМ.
- 1.12. Виконати необхідні попередні розрахунки.
- 1.13. Відповісти на питання самотестування, готовності до виконання роботи.
- 1.14. Для кожного пункту третього розділу проробити та запропонувати свій варіант методики досліджень в загальному вигляді із обґрунтуванням використання конкретної моделюючої програми із пакету прикладних програм. Варіант методики досліджень узгодити із викладачем.
- 1.15. Вибрати тип і структуру моделюючих сигналів та розрахувати конкретні їх параметри і діапазони їх можливих значень. Підготувати програму модель із урахуванням проведених розрахунків та узгодити її з викладачем.
- 1.16. Проаналізувати і запропонувати можливість розширення досліджень стосовно особливостей процедури АЦ-перетворення радіосигналів. Запропонувати варіанти додаткових досліджень та відповідні методики їх проведення в доповнення програми розділу 3.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 17

3. Порядок виконання роботи

3.1. Дослідити вплив у часовій та частотній областях визначення на радіосигнал, що перетворюється в цифрову форму **апертурної невизначеності** (інтервалу невизначеності) дискретизації його у часі. Дослідження виконати для вузькосмугового радіосигналу в залежності від його параметрів (амплітуди, частоти, початкової фази). Дослідження проводити за допомогою програми ADC та Numeri.

Для цього в програмі ADC в меню Сигнал – Синусоїда задаються параметри сигналу (амплітуда, частота та початкова фаза). В меню АЦП – Параметри задаються параметри аналого-цифрового перетворення: розрядність коду квантування R ; граничні значення амплітуди сигналу $[h_{\min}; h_{\max}]$; крок квантування h_{kv} , що визначається в програмі автоматично; період дискретизації T_d ; інтервал невизначеності ΔT_d ; кількість відліків сигналу накопичуваного масиву $N_s = 2^n$.

Отриманий масив відліків можливо проаналізувати у вигляді часової діаграми та таблиці відліків, вибравши на панелі інструментів програми ADC відповідні іконки.

Подальші дослідження необхідно виконувати в частотній області, аналізуючи комплексний спектр сигналу, який визначати за допомогою програми Numeri. Для цього необхідно зберегти отриманий масив відліків сигналу натиснувши іконку (дискета) на панелі інструментів програми ADC, та перенести отриманий файл (10.xu) в папку Numeri. Після запуску програми Numeri вибрати п. 5 (ДПФ), далі натиснути – 1 (Дані), далі – 1 (Створення даних), далі – 2 (дані з файлу на диску), загрузити файл: вводимо – 10 (назва свого файлу) + F10.

Щоб отримати спектр сигналу натиснути тричі Esc, далі – 4 (тип вікна) + 5 (Блекмана), далі – 3 (ДПФ), та оцінити спектр у вигляді таблиці 2 та графіка 1.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 18

3.2. Виконати дослідження пункту 3.1 в залежності від періоду дискретизації.

3.3. Виконати дослідження пункту 3.1 в залежності від форми сигналу у часі та його спектрального складу, а також для полісигнальної адитивної суміші.

3.4. Виконати дослідження пункту 3.1. за умови дискретизації адитивної суміші радіосигналу і білого гаусового шуму, для різних відношень сигнал/шум.

Для цього після запуску програми Numer1 вибрати п. 5 (ДПФ), далі натиснути – 1 (Дані), далі – 1 (Створення даних), далі – 2 (дані з файлу на диску), загрузити файл: вводимо – 10 (назва свого файлу) + F10.

Щоб зашумити дані натиснути двічі Esc, далі – 3 (зашумлення даних), вводимо параметри + F10, далі двічі Esc і отримуємо спектр: 4-5 + 3-1(2)

3.5. Дослідити вплив у часовій та частотній областях визначення на радіосигнал, що перетворюється в цифрову форму, похибки квантування по рівню в залежності від величини **кроку квантування** h_{kv} . Дослідження виконувати для вузько смугового радіосигналу в залежності від його параметрів (амплітуди, частоти, початкової фази).

3.6. Виконати дослідження пункту 3.5 в залежності від форми радіосигналу у часі та його спектрального складу, а також для полісигнальної адитивної суміші.

3.7. Виконати дослідження пункту 3.5 за умови квантування адитивної суміші радіосигналу і білого гаусового шуму для різних відношень сигнал/шум.

3.8. Виконати дослідження пункту 3.5 за умови наявності апертурної невизначеності дискретизації у часі.

3.9. Виконати дослідження пункту 3.8 в залежності від періоду дискретизації радіосигналу.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 19</i>

3.10. Виконати дослідження впливу у часовій та частотній областях визначення процедури АЦ-перетворення адитивної суміші радіосигналу та білого гаусового шуму в залежності від апертурної невизначеності дискретизації у часі та похибки квантування по рівню.

4. Обробка результатів досліджень

4.1. По кожному пункту досліджень розділу 3 виконати якісний аналіз отриманих результатів і виявити основні залежності. Обробку виконати шляхом аналізу часових та спектральних діаграм радіосигналу до і після АЦ-перетворення.

Часові та спектральні реалізації дослідити якісно та кількісно. Якісний аналіз здійснюється визначенням наявності спотворення форми сигналу або його спектра, формуванням паразитних спектральних складових, а також появою широкосмугових спотворень у вигляді шуму.

Кількісний аналіз виконати шляхом визначення рівня паразитних спектральних складових, максимального та середнього значення шуму, ширини спектра і рівня спотворення спектра сигналу (амплітуди, ширини спектра) по відношенню до неспотвореного спектра при ідеальному АЦ-перетворенні, який визначається з використанням програми Numeri (в якій: розрядність АЦП $\gg 1$, і нульовий інтервал апертурної невизначеності).

4.2. По кожному пункту досліджень зняти кількісні характеристики похибок та параметрів радіосигналів і побудувати відповідні графічні залежності (графіки, гістограми, тощо).

4.3. Виконати порівняльний аналіз отриманих результатів досліджень і виявити основні залежності якості впливу параметрів процедури АЦ-перетворення на радіосигнал.

4.4. По кожному пункту виконати теоретичні розрахунки і порівняти їх результати із отриманими експериментальними результатами.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 20</i>

4.5. По результатам виконаних досліджень та розрахунків зробити обґрунтовані аналіз та висновки.

4.6. Результати досліджень оформити у вигляді звіту, що полягає індивідуальному захисту.

Контрольні питання самоконтролю

1. Дати визначення процедури аналог-цифрового перетворення.
2. Дати визначення та навести причину виникнення апертурної невизначеності дискретизації у часі радіосигналу.
3. Дати визначення та навести причину виникнення похибки квантування по рівню при АЦ-перетворенні радіосигналу.
4. Як залежить максимальне та середньоквадратичне значення похибки квантування по рівню радіосигналу від кроку квантування h_a .
5. Як впливає на якість АЦ-перетворення радіосигналу апертурна невизначеність дискретизації у часі?
6. Як розраховується допустимий крок квантування по рівню h_a при АЦ-перетворенні радіосигналу, відеосигналу або одиночного відліку сигналу інформаційного датчика (детектора)?
7. Як розраховується допустима невизначеність дискретизації у часі при АЦ-перетворенні відеосигналу, радіосигналу або одиночного відліку сигналу інформаційного датчика (детектора)?

Рекомендований перелік літератури

1. Proakis J.G. (2006). Digital Signal Processing, Principles, Algorithms, and Applications. 4-th Edition. – New Jersey: Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1077.
2. *Бабак В.П. та ін. Обробка сигналів: Підручник /В.П. Бабак, В.С. Хандецький, Е. Шрюфер. – К: Либідь, 1996. -392с.*

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 21</i>

Практична робота №3

Дослідження методів формування звукових сигналів з використанням МК AVR

1.1. Навчальні питання

1. Розробка принципової схеми музичної шкатулки з використанням мікроконтролера ATtiny2313.
2. Написання програми для музичної шкатулки та моделювання її роботи в середовищі PROTEUS.

1.2. Навчальна мета

1. Практичне ознайомлення з розробкою та програмуванням схем на мікроконтролері ATtiny2313.

1.3. Теоретичні відомості

3.1. Музична шкатулка. Постановка завдання

І так, починаємо ряд прикладів, у яких буде не тільки представлений опис якихось схем на мікроконтролері, але і описаний процес їх розробки. Всі три приклади, наведені в цій книзі спеціально розроблені автором, і призначені для навчання початківців програмістів і є заключним етапом уроку з програмування, докладно представленого в [4].

У першому прикладі ми покажемо процес розробки простого мікропроцесорного пристрою, призначеного для автоматичного відтворення декількох простих мелодій. Причому всі ці мелодії ми жорстко "зашьєм" з вшанування пам'яті мікроконтролера. Зручніше за все для цього використовувати програмну пам'ять контролера. Тобто, ту ж саму пам'ять, куди записується керуюча програма. Можна було б, звичайно, для зберігання мелодій використовувати і EEPROM. Але обсяг EEPROM в мікросхемі ATtiny2313 складає всього 128 байт, а обсяг пам'яті програм - 2 кілобайт. Тому пам'ять програм у даному випадку переважно.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 22

Сформулюємо завдання наступним чином: «Розробити пристрій, призначений для відтворення простих одноголосих мелодій, записаних в пам'ять програм на етапі програмування. Пристрій повинен мати сім кнопок керування. Кожній з кнопок повинна відповідати своя мелодія. Мелодія відтворюється при натисканні та утриманні кнопки. При відпуску всіх кнопок відтворення мелодій припиняється ».

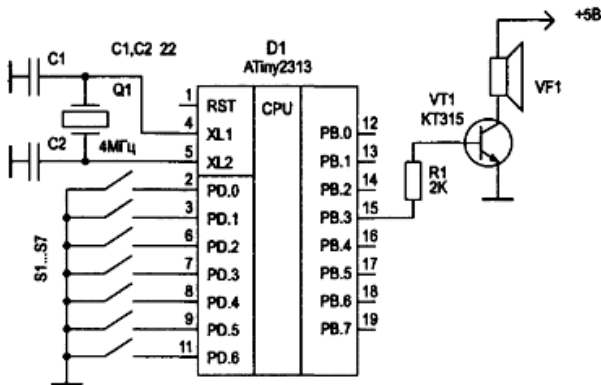


Рис. 3.1. Принципова електрична схема музичної шкалулки
Схема

Принципова електрична схема пристрою, узадовільняюча сформульованим вище вимогам наведена на рис. 3.1. Кнопки S1 ... S7 призначені для вибору мелодій. Для відтворення мелодії використовується звуковий випромінювач VF1, сигнал на який поступає з виходу PB3 мікроконтролера. У як підсилювач сигналу використовується електронний ключ R1, VT1.

Алгоритм

Для початку нам потрібно придумати, як ми будемо зберігати мелодії в пам'яті. Для того, щоб в пам'яті можна було щось зберігати, треба спочатку це щось будь-який спосіб закодувати. Будь-яка мелодія складається з нот. Кожна нота має свій тон (частоту) і тривалість звучання. Для того, щоб закодувати тон ноти, можна

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 23

просто всі ноти пронумерувати по порядку. Зручніше нумерувати, починаючи з найнижчого тону. На клавіатурі клавішного інструменту це буде зліва направо.

Відомо, що весь музичний ряд ділиться на **октави**. Якщо ви думаєте, що в кожній октаві сім нот, то ви погано знаєте фізичні основи музичного ряду. Насправді в сучасному музичному ряду кожна октава ділиться 12 на нот. Сім основних нот (білі клавіші) та п'ять додаткових (чорні клавіші).

Поділ на основні і додаткові ноти склалося історично. В даний час використовується музичний лад, в якому всі 12 нот однієї октави рівнозначні. Частоти будь-яких двох сусідніх нот відрізняються один від одного в однакову кількість разів. При цьому частоти однойменних нот в двох сусідніх октавах відрізняються рівно в два рази. Більш докладно про це ви можете прочитати в [3].

Для нас же важливо те, що коди всіх цих нотах ми повинні привласнювати в порядку зростання частоти, і почнемо ми з ноти «До» першої октави. Для музичної скриньки більш низькі ноти не потрібні. В **табл. 3.1** показані коди для всієї першої октави. Наступна, друга октава продовжує перший і за кодуванням, і по набору частот. Так нота «До» другої октави буде мати код 13, а частоту $f_{12}=f_0 \times 2$. А нота «Ре» другої октави буде мати код 14 і частоту $f_{12}=f_1 \times 2$. І так далі.

Таблиця 3.1.

Кодування нот першої октави

Код	Нота	Частота	Код	Нота	Частота
1	До	f_0	7	Фа#	$f_6=f_5/K$
2	До#	$f_1=f_0/K$	8	Соль	$f_7=f_6/K$
3	Ре	$f_2=f_1/K$	9	Соль#	$f_8=f_7/K$
4	Ре#	$f_3=f_2/K$	10	Ля	$f_9=f_8/K$
5	Ми	$f_4=f_3/K$	11	Ля#	$f_{10}=f_9/K$
6	Фа	$f_5=f_4/K$	12	Си	$f_{11}=f_{10}/K$

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 24</i>

Музична тривалість теж легко кодується. У музиці застосовують не довільну тривалість, а тривалість, виражену частками від цілої (див. табл. 3.2). В залежності від темпу реальна тривалість цілої ноти змінюється. Для збереження мелодії необхідно дотримуватися лише співвідношення між тривалістю. Тому нам необхідно закодувати лише сім варіантів тривалості. Привласнимо їм коди від 0 до 6. Наприклад так, як це показано у графі «Код» табл. 3.2. Призначення графі «Коефіцієнт ділення» ми поки що опустимо.

Таблиця 3.2.

Кодування музичних тривалостей

Код	Длительность	Кoeffициент деления
0	1 (целая)	64
1	1/2 (половинная)	128
2	1/4 (четверть)	256
3	1/8 (восьмая)	512
4	1/16 (шестнадцатая)	1024
5	1/32 (тридцать вторая)	2048
6	1/64 (шестьдесят четвертая)	4096

Крім нот, будь-яка мелодія обов'язково містить **музичні паузи**.

Визначення. *Паузи* - це проміжки часу, коли ні один звук не звучить. Загальна тривалість музичних пауз приймає точно такі ж значення, як і тривалість нот.

У зв'язку з цим зручно представити паузу як ще одну ноту. Ноту без звуку. Такий ноті логічно привласнити нульовий код.

Кодуємо мелодії

Для економії пам'яті зручніше кожен ноту кодувати одним байтом. Домовимося, що три старших біта ми будемо використовувати для кодування тривалості ноти, а решта п'ять бітів - для кодування її тону. П'ятьма бітами можна закодувати до 32 різних нот, що цілком вистачить для музичної скриньки.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 25</i>

Отже, якщо використовувати наведений вище спосіб кодування, то код ноти ля першої октави тривалістю 1 / 4 в двійковому вигляді буде рівний.

$$\underline{01001001} = 73$$

Тепер ми можемо приступати до **кодування мелодій**. Для того, щоб закодувати мелодію, нам потрібна її нотний запис. Використовуючи нотний запис, ми повинні привласнити кожній ноті і кожної музичної паузи свій **код**.

Ланцюжок таких кодів і буде являти собою закодовану мелодію. За умовами завдання наша музична скриня повинна вміти відтворювати сім різних мелодій. Коди всіх семи мелодій ми розмістимо в програмній пам'яті мікроконтролера.

Як визначити **кінець кожній мелодії**? Для того, щоб комп'ютер знав, де закінчується кожна мелодія, використовуємо код 255 як ознака кінця.

Тепер нам потрібно придумати, як мікроконтроллер буде знаходити **початок кожної мелодії**. Всі мелодії мають різну довжину, а в пам'яті вони будуть записані одна за одною. Тому адресу початку кожної мелодії залежить від довжини всіх попередніх. Зручніше за все просто за фактом визначити адресу початку кожної мелодії і помістити всі сім адрес в спеціальну таблицю.

Крім цієї таблиці нам ще знадобиться таблиця коефіцієнтів розподілу для всіх 32 нот і таблиця, що зберігає константи затримки для всіх використовуваних нами музичних тривалостей.

Алгоритм роботи музичної скриньки

Тепер ми можемо сформулювати алгоритм роботи музичної скриньки.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 26</i>

1. Просканувати клавіатуру і визначити номер найпершої натиснутою клавіші.
2. Витягти з таблиці почав мелодій значення елемента, номер якого відповідає тільки що певним номером натиснутою клавіші. Це значення дорівнюватиме адресою в програмній пам'яті, де починається потрібна нам мелодія.
3. Почати цикл відтворення мелодії. Для цього по черзі витягати коди нот з пам'яті, починаючи з адреси, на яку ми визначили в пункті 2 алгоритму.
4. Кожен код ноти розкласти на код тону і код тривалості.
5. Якщо код тону дорівнює нулю, вимкнути звук і перейти до формування затримки (до п. 9 цього алгоритму).
6. Якщо код тону не дорівнює нулю, витягти з таблиці коефіцієнтів ділення значення елемента з номером, рівним коду тону.
7. Записати коефіцієнт розподілу, який ми знайшли в пункті 6 цього алгоритму, у реєстр збігу таймера T1.
8. Увімкнути звук (підключити висновок ОС1А до виходу таймера T1).
9. Витягти з таблиці тривалостей затримки значення елемента з номером, рівним коду тривалості.
10. Сформуванати паузу з використанням константи затримки, яку ми знайшли в пункті 9 цього алгоритму.
11. Після закінчення паузи вимкнути звук (відключити ОС1А від виходу таймера),
12. Повторювати цикл (пункти 4-11 цього алгоритму) до тих пір, поки натиснута відповідна кнопка.
13. Якщо черговий код ноти виявиться рівним 255, перейти на початок поточного мелодії, тобто повернутися до п. 3 цього алгоритму.

Програма на Асемблері

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 27</i>

Можливий варіант програми на мові Асемблер приведений в лістингу 3.1. Приводиться нижче опис програми розраховано на читачів, знайомих з мовою Асемблера. Якщо ви не знаєте цієї мови, рекомендую звернутися до [4], де представлений докладний урок для початківців.

Лістинг 3.1

```
#####
##                Пример 9                ##
:##            Музыкальная шкатулка        ##
#####

----- Псевдокоманды управления

1  include "tn2313def.inc"                ; Присоединение файла описаний
2  .list                                  ; Включение листинга

3  .def      loop1 = R0                    ; Три ячейки для процедуры задержки
4  .def      loop2 = R1
5  .def      loop3 = R21
6  .def      temp = R16
7  .def      count = R17
8  .def      fnota = R19
9  .def      dnota = R20
10 .def

----- Начало программного кода

11      cseg                                ; Выбор сегмента программного кода
12      org      0                          ; Установка текущего адреса на ноль

13  start:  rjmp   init                    ; Переход на начало программы
14          reti   0                        ; Внешнее прерывание 0
15          reti   !                        ; Внешнее прерывание 1
16          reti   1                        ; Таймер/счетчик 1 захват
17          reti   1                        ; Таймер/счетчик 1. совпадение канал А
18          reti   1                        ; Таймер/счетчик 1. прерывание по переполнению
19          reti   0                        ; Таймер/счетчик 0, прерывание по переполнению
20          reti   0                        ; Прерывание UART прием завершен
21          reti   0                        ; Прерывание UART регистр данных пуст
22          reti   0                        ; Прерывание UART передача завершенна
23          reti   0                        ; Прерывание по компаратору
24          reti   0                        ; Прерывание по изменению на любом контакте
25          reti   1                        ; Таймер/счетчик 1 Совпадение, канал В
26          reti   0                        ; Таймер/счетчик 0 Совпадение, канал В
27          reti   0                        ; Таймер/счетчик 0 Совпадение, канал А
28          reti   0                        ; Прерывание по готовности к старту
29          reti   0                        ; USI Переполнение
30          reti   0                        ; EEPROM Готовность
31          reti   0                        ; Переполнение охранного таймера
```

 * Модуль инициализации *

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 28

```

init
----- Инициализация стека
32          id1      temp. RAMEND      . Инициализация стека
33          out      SPL temp
----- Инициализация портов В/В
34          id1      temp. 0x08      . Инициализация порта PB
35          out      PORTB temp
36          out      DDRB temp
35          id1      temp. 0x7F      . Инициализация порта PD
36          out      PORTD temp
37          id1      temp. 0x00
38          out      DDRD temp
----- Инициализация (выключение) компаратора
39          id1      temp. 0x80
40          out      ACSR temp
----- Инициализация таймера T1
41          id1      temp. 0x09      . Включаем режим CTC
42          out      TCCR1B temp
43          id1      temp. 0x00      . Выключаем звук
44          out      TCCR1A temp
-----
. *          Начало основной программы          *
-----
main:
----- Вычисление номера нажатой кнопки

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 29</i>

45		clr	count	: Обнуление счетчика опроса клавиш
46		in	temp, PIND	: Чтение порта D
47	m2.	lsl	temp	: Сдвигаем входной байт
48		brc	m3	: Если текущий разряд был равен 0
49		inc	count	: Увеличиваем показание счетчика
50		spi	count, 7	: Сравнение (7 - конец сканирования)
51		brc	m2	: Если не конец, продолжить
52		rjmp	m1	: Если не одна клавиша не нажата
----- Выбор мелодии				
53	m3.	mov	YL, count	: Вычисляем адрес где
54		ldi	ZL, low(tabm*2)	: хранится начало мелодии
55		ldi	ZH, high(tabm*2)	
56		rcall	addrw	: К подпрограмме 16-разрядного сложения
57		lpm	XL, Z+	: Извлекаем адреса из таблицы
58		lpm	XH, Z	: и помещаем в X
----- Воспроизведение мелодии				
59	m4	mov	ZH, XH	: Записываем в Z начало мелодии
60		mov	ZL, XL	
61	m5	in	temp, PIND	: Читаем содержимое порт D
62		spi	temp, 0x7F	: Проверяем на равенство 7FH
63		breq	m1	: Если равно (кнопки отпущены) в начало
64		lpm	temp, Z	: Извлекаем код ноты
65		spi	temp, 0xFF	: Проверяем не конец ли мелодии
66		breu	m4	: Если конец начинаем мелодию сначала
67		and	temp, 0x1F	: Выделяем код тона из кода ноты
68		mov	rnota, temp	: Записываем в регистр кода тона
69		lpm	temp, Z+	: Еще раз берем код ноты
70		rol	temp	: Производим четырехкратный сдвиг кода ноты
71		rol	temp	
72		rol	temp	
73		rol	temp	

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 30

```

74      andl    temp, 0x07      . Выведем код длительности
75      mov     dnota, temp     . Помещаем ее в регистр длительности
76      rcall   nota          . К подпрограмме воспроизведения ноты
77      rjmp   m5             . В начало цикла (следующая нота)

```

```

.....
:~      Вспомогательные подпрограммы      ~
.....

```

```

----- Подпрограмма 16-ти разрядного сложения
78      addw   push    YH
79      lsl    YL          . Умножение первого слагаемого на 2
80      ldl    YH, 0      . Второй байт первого слагаемого = 0
81      add   ZL, YL      . Складываем два слагаемых
82      adc   ZH, YH
83      pop   YH
84      ret

----- Подпрограмма исполнения одной ноты
85      nota.  push    ZH
86      push  ZL
87      push  YL
88      push  temp
89      cpl   fnota, 0x00 . Проверка, не пауза ли
90      breq  nt1        . Если пауза, переходим сразу к задержке
91      mov   YL, fnota   . Вычисляем адрес, где хранится
92      ldi   ZL, low(tabkd*2) . коэффициент деления для текущей ноты
93      ldi   ZH, high(tabkd*2)
94      rcall addw        . К подпрограмме 16-разрядного сложения

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 31

```

95      lpm      temp, Z+      ; Извлекаем мл. разряд КД для текущей ноты
96      lpm      temp1, Z      ; Извлекаем ст. разряд КД для текущей ноты
97      out      OCR1AH, temp1 ; Записать в старш. часть регистра совпадения
98      out      OCR1AL, temp  ; Записать в младш. часть регистра совпадения

99      ldi      temp, 0x40    ; Включить звук
100     out      TCCR1A, temp

101     nt1:    rcall     wait      ; К подпрограмме задержки

102     ldi      temp, 0x00    ; Выключить звук
103     out      TCCR1A, temp

104     ldi      dnota, 0      ; Сбрасываем задержку для паузы между нотами
105     rcall     wait          ; Пауза между нотами

106     pop      temp          ; Завершение подпрограммы
107     pop      YL
108     pop      ZL
109     pop      ZH
110     ret

:----- Подпрограмма формирования задержки
111     wait:    push     ZH
112             push     ZL
113             push     YH
114             push     YL

115     mov      YL, dnota      ; Вычисляем адрес где хранится
116     ldi      ZL, low(tabz*2) ; нужный коэффициент задержки
117     ldi      ZH, high(tabz*2)
118     rcall     addw          ; К подпрограмме 16-разрядного сложения

119     lpm      YL, Z+        ; Читаем первый байт коэффициента задержки
120     lpm      YH, Z         ; Читаем второй байт коэффициента задержки

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015		Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1		Арк 120 / 32

```

121      clr      ZL      . Обнуляем регистровую пару Z
122      clr      ZH

      . Цикл задержки
123  w1      id1      loop, 255      : Пустой внутренний цикл
124  w2      dec      loop
125      brne     w2
126      adiw     R30, 1      . Увеличение регистровой пары Z на единицу
127      cp      YL, ZL      : Проверка младшего разряда
128      brle     w1
129      cp      YH, ZH      : Проверка старшего разряда
130      brne     w1

131      pop      YL      : Завершение подпрограммы
132      pop      YH
133      pop      ZL
134      pop      ZH
135      ret

```

```

:.....
: *      Таблица длительности задержек      *
:.....

```

```

136  tabz:      dw      128, 256, 512, 1024, 2048, 4096, 8192

```

```

:.....
: *      Таблица коэффициентов деления      *
:.....

```

```

137  tabkd:     .dw      0
138      .dw      4748, 4480, 4228, 3992, 3768, 3556, 3356, 3168, 2990, 2822, 2664, 2514
139      .dw      2374, 2240, 2114, 1996, 1884, 1778, 1678, 1584, 1495, 1411, 1332, 1257
140      .dw      1187, 1120, 1057, 998, 942, 889, 839, 792

```

```

:.....

```


Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 33

.* Таблиця начал всех мелодий *

141 tabl: dw mel1*2, mel2*2, mel3*2, mel4*2
142 .dw mel5*2, mel6*2, mel7*2

.* Таблица мелодий *

: В траве сидел кузнечик
143 mel1: .db 109, 104, 109, 104, 109, 108, 108, 96, 108, 104
144 .db 108, 104, 108, 109, 109, 96, 109, 104, 109, 104
145 .db 109, 108, 108, 96, 108, 104, 108, 104, 108, 141
146 .db 96, 109, 111, 79, 79, 111, 111, 112, 80, 80
147 .db 112, 112, 112, 111, 109, 108, 109, 109, 96, 109
148 .db 111, 79, 79, 111, 111, 112, 80, 80, 112, 112
149 .db 112, 111, 109, 108, 141, 128, 96, 255

: Песенка крокодила Гены
150 mel2: .db 109, 110, 141, 102, 104, 105, 102, 109, 110, 141
151 .db 104, 105, 107, 104, 109, 110, 141, 104, 105, 139
152 .db 109, 110, 173, 96, 114, 115, 146, 109, 110, 112
153 .db 109, 114, 115, 146, 107, 109, 110, 114, 112, 110
154 .db 146, 109, 105, 136, 107, 105, 134, 128, 128, 102
155 .db 105, 137, 136, 128, 104, 107, 139, 137, 128, 105
156 .db 109, 141, 139, 128, 110, 109, 176, 112, 108, 109
157 .db 112, 144, 142, 128, 107, 110, 142, 141, 128, 105
158 .db 109, 139, 128, 173, 134, 128, 128, 109, 112, 144
159 .db 142, 128, 107, 110, 142, 141, 128, 105, 109, 139
160 .db 128, 173, 146, 128, 96, 255

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 34</i>

		В лесу родилась елочка
161	mel3	db 132. 141. 141. 139. 141. 137 132. 132. 132. 141
162		db 141. 142. 139. 176. 128. 144. 146. 146. 154. 154
163		db 153. 151. 149. 144. 153. 153. 151. 153 181. 128
164		db 96 255
		Happy births to you
165	mel4	db 107. 107. 141. 139 144. 143. 128. 107. 107. 141
166		db 139. 146. 144. 128 107. 107. 151. 148. 146 112
167		db 111. 149. 117. 117 148. 144. 146. 144. 128 255
		С чего начинается родина
168	mel5	db 99. 175. 109. 107. 106. 102. 99. 144. 111. 175
169		db 96. 99. 107. 107. 107. 107. 102. 104. 170. 96
170		db 99. 109. 109. 109. 109. 107. 106. 143. 109. 141
171		.ac 99. 109. 109. 109. 109. 104. 106. 171. 96. 99
172		db 111. 109 107. 106. 102. 99. 144. 111. 143 104
173		db 114. 114. 114. 114. 109. 111. 176. 96. 104. 116
174		db 112. 109 107. 106. 64. 73. 143. 107. 131. 99
175		db 144. 80 80. 112. 111. 64. 75. 173. 128. 255
		Песня из кинофильма Веселье ребята
176	mel6:	.db 105. 109. 112. 149. 116. 64. 80. 148. 114. 64
177		.db 78. 146 112. 96. 105. 105. 109. 144. 111. 64
178		.db 80 145. 112. 64. 81. 178. 96. 117. 117. 117
179		.db 149. 116. 64. 82. 146. 112. 64. 79. 146. 144
180		.db 96. 105. 105. 107. 141. 108. 109. 112. 110. 102
181		.db 104. 137. 128. 96. 105. 105. 105. 137. 102. 64
182		.db 73. 142. 105. 107. 109 64. 75. 137. 96. 105
183		db 105. 105 137. 102. 105. 142. 112. 64. 82. 180
184		db 96. 116. 116. 116. 148. 114. 112. 142. 109. 64
185		db 78. 146. 144. 96. 105. 105. 107. 141. 108. 109
186		db 112. 110. 102. 104. 169. 96. 96. 255

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 35

	;	Улюбка	
187	mel7	db	107. 104. 141. 139. 102. 105. 104. 102. 164. 128
188		db	104. 107. 109. 109. 109. 111. 114. 112. 111. 109
189		db	144. 139. 128. 109. 111. 144. 96. 111. 109. 104
190		db	107. 105. 173. 128. 111. 109. 112. 107. 111. 109
191		db	109. 107. 102. 104. 134. 132. 128. 100. 103. 107
192		db	107. 107. 107. 139. 112. 100. 103. 102. 102. 102
193		db	134. 102. 103. 107. 105. 107. 108. 108. 108. 108
194		db	107. 105. 107. 108. 144. 142. 128. 112. 107. 110
195		db	140. 112. 105. 108. 107. 107. 107. 105. 140. 139
196		db	139. 112. 103. 102. 103. 105. 108. 107. 105. 103
197		db	128. 112. 107. 110. 108. 108. 108. 140. 112. 105
198		db	108. 107. 107. 107. 139. 112. 103. 102. 103. 105
199		db	108. 107. 105. 103. 105. 139. 132. 128. 96. 96
200		db	96. 255

Опис програми (лістинг 3.1)

Опис програми зручніше почати з кінця. Починаючи з рядка 136 програми розташовується опис так званих таблиць даних. Насправді кожна із цих "таблиць" представляє собою ланцюжок кодів, записуваних у програмну пам'ять мікроконтролера й призначених для кодування того або іншого виду даних. Для опису цих даних використовуються як оператори db, так і оператори dw.

Перша таблиця містить коефіцієнти затримки для формування всіх варіантів музичної тривалості. Таблиця починається з адреси, відповідного до мітки tabz. Уся таблиця займає один **рядок** програми (**рядок 136**). Тому що в нашій програмі ми будемо застосовувати лише сім варіантів тривалості, таблиця має 7 елементів. Кожний елемент записується на згадку як двухбайтне слово.

У **рядках 137–140** описується **таблиця коефіцієнтів розподілу для Всіх нот**. Початок таблиці відповідає мітці tabkd. Кожний елемент цієї таблиці також має розмір у два байти. Перший елемент таблиці дорівнює нулю. Це невикористований елемент.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 36</i>

Ноти номер нуль у нас не існує. Нуль ми використовували для кодування паузи.

У паузі не формується звуковий сигнал, тому й коефіцієнт розподілу там не має значення. Тому значення нульового елемента масиву несуттєво. Опис таблиці розбитий на рядки. Для зручності кожний рядок описує коефіцієнти розподілу для однієї октави. Нульова нота виділена в окремий рядок. Остання октава неповна, тому що наша скринька буде використовувати всього 32 ноти.

У рядках 143–200 описана **таблиця мелодій**. Вірніше, це не одна таблиця, а сім таблиць (своя таблиця для кожної з мелодій). Кожна таблиця позначена своєю окремою міткою (ті 11, ті 12 – ті 17). Значення кожної мітки – це адреса початку відповідної до мелодії. Кожне значення таблиці мелодій записується на згадку у вигляді одного байта. Тому всі рядки, крім останньої, для кожної таблиці мають парне число значень.

У рядках 141,142 описана **таблиця початків усіх мелодій**. Початок цієї таблиці відзначене міткою `tabm`. Таблиця використовується для того, щоб програма могла знайти адресу початку потрібної мелодії по її номеру. У якості елементів масиву виступають подвоєні значення міток `mel1, mel2 – mel7`. Застосування подвоєних значень обумовлене необхідністю перекладу адрес із основної адресації в альтернативну. При трансляції програми замість міток на згадку будуть записані конкретні адреси.

Докладніше про основну й альтернативну адресацію читайте в [3] і [4].

Процедура обчислення адреси

Велика кількість таблиць у нашій програмі змушує подбати про процедуру обчислення адреси.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 37</i>

Однотипні обчислення зручно оформити у вигляді підпрограми. Ця підпрограма займає **рядки 78–84**. Виклик підпрограми проводиться по імені `addw`. Підпрограма одержує номер елемента таблиці й адреса її початку. Номер елемента передається в підпрограму за допомогою реєстру `YL`, а адреса – через реєстрову пару `Z`.

Використовуючи ці дані, підпрограма обчислює **адресу потрібного елемента**. Для цього вона спочатку подвоєє номер елемента (**рядок 79**). Потім доповнює отримане значення до шістнадцятирозрядну шляхом запису в `YN` нульового байта (**рядок 80**). І, нарешті, робить додавання двох шістнадцятирозрядних величин, що перебувають до цього моменту в реєстрових парах `Y` і `Z` (**рядка 81, 82**). Результат обчислень при цьому попадає, у реєстрову пару `Z`. Призначення команд `push` і `pop` (**рядка 78, 83**), очевидно, уже пояснювати не потрібно.

Текст програми «крок за кроком»

Тепер розглянемо текст програми із самого початку. У рядках 3...10 розташований модуль опису змінних (робочих реєстрів). У рядках 13...31 розташовується модуль перевизначення векторів переривань, у рядках 32...41 – модуль команд ініціалізації.

Особливості програми

Процедура, розташована в **рядках 45–52** програми, сканує клавіатуру й знаходить код першої з натиснутих кнопок. Найдений код перебуває в реєстрі `count`. Потім керування переходить до **рядка 53**. Із цього місця починається процедура вибору мелодії (**рядка 53–58**). Суть процедури – прочитати з таблиці `tabm` значення адреси початку цієї мелодії. Тобто прочитати елемент таблиці, номер якого дорівнює коду натиснутої клавіші.

Перш ніж прочитати елемент, необхідно **знайти його адресу**. Для обчислення адреси використовуємо підпрограму `addw`. Перед тем, як викликати підпрограму, підготуємо всі дані. Номер натиснутої клавіші поміщаємо в реєстр `YL` (**рядок 53**). Адреса початку таблиці

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 38</i>

записуємо в реєстрову пару Z (рядка 54, 55). І лише потім у рядку 56 викликається підпрограма `addw`.

Після виходу з підпрограми в реєстровій паре Z перебуває результат обчислень – адреса потрібного нам елемента таблиці `tabm`. Наступні дві команди (рядка 57 і 58) витягають той елемент (адреса початку мелодії) і поміщають його в реєстрову пару X. Там ця адреса буде зберігатися увесь час, поки відтворюється саме ця мелодія.

Наступний етап – відтворення мелодії. Відтворенням мелодії займається процедура, розташована в рядках 59–77. Для послідовного відтворення нот нам знадобиться покажчик поточної ноти. У якості покажчика поточної ноти використовується реєстрова пара Z. На самому початку процедури відтворення мелодії в реєстрову пару Z міститься адреса початку мелодії їх реєстрової пари X (рядка 59, 60).

Потім починається цикл відтворення (рядка 61–77). У цьому циклі програма витягає код ноти за адресою, на яку вказує наш покажчик, виділяє з коду ноти код тону й код тривалості, відтворює ноту, а потім збільшує значення покажчика на одиницю. Потім увесь цикл повторюється.

Цей процес відбувається доти, поки код чергової ноти не виявиться рівним 255 (мітка кінця мелодії). Прочитавши цей код, програма передає керування на рядок 62, де в реєстр Z знову записується адреса початку мелодії. Відтворення мелодії почнеться спочатку. Цей процес повинен перерватися лише в одному випадку – при відпусканні керуючої кнопки.

Для перевірки стану кнопок у цикл відтворення мелодії включена спеціальна процедура (рядка 61–63). Процедура спрощено перевіряє стан відразу всіх кнопок. Вона зчитує вміст порту PD (рядок 61) і порівнює його з кодом 0x7F (рядок 62). Прочитане з порту значення може бути рівно 0x7F тільки в одному випадку – якщо

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 39</i>

всі кнопки відпущенні. Якщо хоча б одна кнопка натиснута, то при читанні порту ми одержимо інше значення.

Перевіркою вищеописаної умови займається оператор `breq` у **рядку 63**. Якщо всі кнопки виявилися відпущені, цей оператор завершує цикл відтворення мелодії й передає керування на мітку `ml`, тобто на самий початок основного циклу програми. Там відбувається вимикання звуку, а потім нове сканування клавіатури.

Якщо хоча б одна кнопка виявиться натиснутою, то цикл відтворення звуку триває далі, і керування переходить до **рядка 64**, де відбувається добування коду ноти. Тому що адреса цієї ноти перебуває в реєстровій парі `Z` (покажчик поточної ноти), те для добування ноти просто використовується команда `lpm`.

У **рядку 65** відбувається перевірка ознаки кінця мелодії. Тільки що прочитаний код ноти рівняється з кодом `0xff`. Оператор `breq` у **рядку 66** передає керування по мітці `t4`, якщо мелодія дійсно закінчилася (умова виконується). Якщо код ноти не рівний `0xff`, переходу не відбувається, і керування переходить до **рядка 67**.

У **рядках 67–75** відбувається обробка коду ноти. Тобто з коду ноти виділяється код тону й код тривалості. Спочатку на код ноти накладається маска, яка залишає п'ять молодших розрядів, а три старші скидає (**рядок 67**). Під дією маски в реєстрі `temp` залишається код тону, який потім міститься в реєстрі `fnota` (**рядок 68**).

Тепер нам потрібно знайти **код тривалості ноти**. Для цього нам заново прийде витягти код ноти з пам'яті програм. Тому що до цього моменту ми не змінювали положення покажчика поточної ноти, то для добування немає ніяких перешкод. У **рядку 69** ми повторно витягаємо код ноти з пам'яті програм. Але цього разу значення покажчика збільшується. Тепер можна приступати до виділення коду тривалості. Як ви пам'ятаєте, тривалість кодується трьома молодшими бітами коду ноти. Для виділення цих бітів нам також потрібно використовувати **маску**. Але однієї маскою нам не обійтися.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 40

Нам потрібно не просто виділити три старші розряди, а зробити їхніми молодшими, як це показано на **рис. 3.2**.

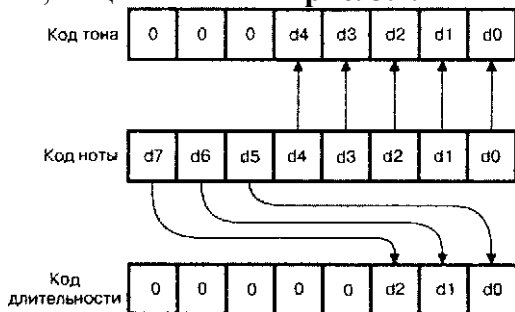


Рис. 3.2. Розкладання коду ноти

Процедура виділення коду тривалості займає **рядки 70–74**.

Спочатку програма робить багаторазове циклічне зрушення коду ноти доти, поки три старші розряди не стануть трьома молодшими. Для зрушення використовується команда `rol`. Тому що зрушення відбувається через гніздо ознаки переносу, то нам знадобиться чотири команди зрушення. Ці команди займають у програмі **рядка 70–73**.

Потім у **рядку 74** на отримане в результаті зрушень число накладається маска, яка виділяє три молодші біта, а п'ять старших скидає в нуль. Отриманий у такий спосіб код тривалості записується в регістр `dnote` (**рядок 75**).

Коли **код тону** й **код тривалості** визначені, проводиться виклик підпрограми відтворення ноти (**рядок 76**). Оператор `gjmp` у **рядку 77** передає керування на початок циклу відтворення мелодії, і цикл повторюється для наступної ноти.

Підпрограма відтворення ноти займає **рядки 85–110**. Вона виконує наступні дії:

- витягає з таблиці `tabkd` коефіцієнт розподілу, відповідний до коду ноти;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 41</i>

- програмує таймер і включає звук;
- потім витримує паузу й звук виключає.

Якщо код тону дорівнює нулю (потрібно відтворити паузу без звуку), добування коефіцієнта розподілу й включення звуку не виконується. Підпрограма відразу переходить до формування паузи.

Починається підпрограма відтворення ноти зі збереження всіх використовуваних реєстрів (**рядка 85–88**). Потім проводиться перевірка коду ноти на рівність нулю (**рядок 89**). Якщо код ноти дорівнює нулю, то оператор `breq` у **рядку 90** передає керування по мітці `ntl`, тобто до рядка, де відбувається виклик процедури формування затримки.

Якщо код ноти не дорівнює нулю, то програма приступає до добування коефіцієнта розподілу. Для обчислення адреси елемента таблиці `tabkd`, де перебуває цей коефіцієнт, знову використовується підпрограма `addw`.

Код тону міститься в реєстр `YL` (**рядок 91**), а адреса початку таблиці – у реєстрову пару `Z` (**рядка 92, 93**). Виклик підпрограми `addw` проводиться в **рядку 94**. У реєстровій парі `Z` підпрограма повертає адресу елемента таблиці, де перебуває потрібний нам коефіцієнт розподілу. У **рядках 95, 96** з таблиці дістається цей коефіцієнт. А в **рядках 97,98** він міститься в реєстр збігу таймера. У **рядках 99,100** включається звук.

У **рядку 104** викликається спеціальна підпрограма, призначена для формування затримки. Підпрограма називається `wait` і формує затримку зі змінною тривалістю. Тривалість затримки залежить від значення реєстру `dnota`. По закінченню затримки звук вимикається (**рядка 102,103**).

На цьому можна було б закінчити процес відтворення ноти. Однак це ще не все. Для правильного звучання мелодії між двома сусідніми нотами необхідно забезпечити хоча б **невелику паузу**. Якщо такої паузи не буде, ноти будуть звучати разом. Це спотворить

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 42

мелодію, особливо якщо підряд іде кілька нот з однаковим тоном. Формування паузи між нотами відбувається в **рядках 104,105**. Допоміжна пауза формується за допомогою вже знайомий нам підпрограми затримки. У **рядку 104** коду паузи привласнюється нульове значення (вибирається сама мінімальна пауза). Потім у **рядку 105** викликається підпрограма wait. Після закінчення паузи залишається тільки відновити вміст усіх збережених реєстрів зі стека (рядок 106–109) і вийти з підпрограми (рядок 110).

Підпрограма формування затримки

І останнє, що нам ще залишилося розглянути, – це підпрограма формування затримки. Текст підпрограми займає **рядки 111–135**. Як і будь-яка інша підпрограма, підпрограма wait на початку зберігає (**рядка 111–114**), а наприкінці – відновлює (**рядка 131–134**) усі використовувані реєстри.

Розглянемо, як працює ця підпрограма. Спочатку визначається тривалість затримки. Для цього дістається відповідний елемент із таблиці tabz. Номер елемента відповідає коду затримки, що перебуває в реєстрі dnota. Добування значення з таблиці проводиться вже знайомим нам образом. Команди, що реалізують обчислення адреси потрібного елемента таблиці, перебувають у **рядках 115–118**. Потім у **рядках 119 і 120** проводиться читання елемента таблиці. Прочитаний код затримки міститься в реєстрову пару Y.

Тепер **наше завдання: сформувати затримку, пропорційну вмісту реєстрової пари Y**. Тому що мікроконтролер Atiny2313 має тільки один шістнадцятирозрядний таймер, який уже зайнятий формуванням звуку, будемо формувати затримку програмним шляхом. Але в цьому випадку цикл формування затримки побудований небагато по-іншому.

Взагалі, способів побудови подібних підпрограм може бути нескінченна безліч. Усе залежить від винахідливості програміста.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 43</i>

Використаний у даному прикладі спосіб більш зручний для формування затримки змінної тривалості, пропорційної заданому коефіцієнту. Головною **особливістю** нового способу є шістнадцятирозрядний параметр циклу.

Для зберігання цього параметра використовується реєстрова пара Z . Перед початком циклу затримки в неї записується нуль. Потім починається цикл, на кожному проході якого вміст реєстрової пари Z збільшується на одиницю. Після кожного такого збільшення проводиться порівняння нового значення Z із вмістом реєстрової пари Y .

Закінчується цикл тоді, коли вміст Z і вміст Y виявляться рівні. У результаті число, записане в реєстровій парі Y , буде визначати кількість проходів циклу. Тому й час затримки, сформоване цим циклом, буде пропорційно константі затримки. Однак цей час буде занадто мало для одержання прийняттого темпу відтворення мелодій. Для того, щоб збільшити час до потрібної нам величини, усередину головного циклу затримки поміщений ще один цикл, що має фіксоване кількість проходів.

Описана вище процедура затримки займає **рядки 121–135**. У **рядках 121, 122** проводиться запис нульового знамення в реєстрову пару Z . Великий цикл затримки займає **рядки 123–130**. Малий внутрішній цикл займає **рядки 124–125**. Для зберігання параметра малого циклу використовується реєстр $loop$. У **рядку 123** у нього записується початкове значення. **Рядка 124,125** виконуються доти, поки вміст $loop$ не виявиться рівним нулю.

У **рядку 126** вміст реєстрової пари Z збільшується на одиницю. У **рядках 127–130** проводиться порівняння вмісту двох реєстрових пар Y і Z . Порівняння проводиться побайтово. Спочатку рівняються молодші байти (**рядок 127**). Якщо вони не рівні, оператор умовного переходу в **рядку 128** передає керування на початок циклу.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 44</i>

Якщо молодші байти рівні, рівняються старші байти (**рядок 129**). Якщо старші байти неоднакові, оператор `brne` у **рядку 130** знову змушує цикл починатися з початку. І тільки коли обоє оператора порівняння дадуть позитивний результат (не викличуть переходу), цикл закінчується, і підпрограма формування затримки переходить до завершальної фази (до **рядків 131–135**).

2.4. Порядок виконання роботи

1. Створити новий проект в середовищі PROTEUS. Скласти схему музичної шкатулки згідно рис. 3.1.

2. Створити новий проект в AVR Studio, вибрати New Project, далі вибрати як налагоджувальну платформу **Atmel AVR Assembler, AVR Simulator** і мікросхему **ATtiny2313**.

3. Для мікроконтролера ATtiny2313 написати програму музичної шкатулки на дві мелодії. Вибрана мелодія повинна грати при натисканні кнопки. Варіант мелодій задає викладач.

Для створення програми користуватися інформацією з теоретичних відомостей та лістингом шаблону 3.1.

4. Виконати відладку програму в AVR Studio в покроковому режимі за допомогою миші або кнопки **F11**.

5. Відкрити створений в PROTEUS проект і підключити написану програму до мікропроцесора. Запустити симуляцію, перевірити працездатність схеми.

6. На основі першої програми створити музичну шкатулку у якій мелодії будуть звучати в іншій тональності – на октаву нижче.

7. Відкрити створений в PROTEUS проект і підключити написану програму до мікропроцесора. Запустити симуляцію, перевірити працездатність схеми.

8. Написати звіт. Зробити висновки.

1.5. Зміст звіту. Звіт повинен містити:

1. Тексти написаних програм.

2. Перелік використаних команд з поясненням їх призначення.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 45</i>

3. Зробити висновки: про об'єм виконаної роботи, чи досягнена мета роботи, чи практичні результати співпали з теоретичними.

1.6. Контрольні тестові питання.

1. Які мікроконтролери підтримує емулятор електронних пристроїв PROTEUS?

2. Що відображає панель **DEVICES**?

3. Як виконати подачу живлення або певного сигналу в довільну точку схеми?

4. Який вигляд мають файли прошивки виконані на мові с, assembler, та в машинних кодах?

5. Які дії необхідно провести, щоб проемувати в PROTEUS роботу мікроконтролера?

6. Як задати частоту тактування МК при емуляції?

7. Яке призначення має віртуальний термінал?

8. Як задати необхідний час емуляції?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 46

Практична робота №4

Дослідження методів опізнавання сигналів з використанням мікроконтролерів AVR серії ATtiny

1.1. Навчальні питання

1. Розробка принципової схеми кодового замка з використанням мікроконтролера ATtiny2313.
2. Написання програми для кодового замка та моделювання її роботи в середовищі PROTEUS.

1.2. Навчальна мета

1. Практичне ознайомлення з розробкою та програмуванням схем на мікроконтролері ATtiny2313.

1.3. Теоретичні відомості

3.1. Кодовий замок. Постановка завдання

У якості прикладу підбрано завдання досить складне й цікаве, здатне навчити працювати із ще неохопленими елементами мікроконтролера. Зручним прикладом є **кодовий замок**. Взагалі, мікроконтролери AVR з їхньою вбудованою енергонезалежною пам'яттю (EEPROM) дають широкий простір для розробника подібних конструкцій. Пам'ять EEPROM ідеально підходить для зберігання коду. Причому такий код завжди легко поміняти.

При розробці замка запропоновано не зовсім звичайний замок. Це кодовий замок, який може сприймати в якості кодової комбінації кнопки, що не тільки окремо натискаються, але й будь-які їхні комбінації. Наприклад кнопки, що попарно натискаються, комбінації типу «Нажати кнопку 6 і, не відпускаючи, набрати код 257». І взагалі, вибрати будь-яку комбінацію будь-яких кнопок у будь-якій комбінації.

Принцип дії замка наступний: у режимі запису коду власник натискає кнопки набору коду в будь-якому порядку й у будь-яких комбінаціях. Мікроконтролер відслідковує всі зміни на клавіатурі й

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 47</i>

записує їх в ОЗП. Довжина кодової послідовності обмежена тільки розмірами ОЗП. Сигналом до закінчення введення коду служить припинення маніпуляцій із клавіатурою.

Вважається, що маніпуляції закінчилися, якщо стан клавіатури не змінився протягом контрольного проміжку часу. Його вибрано приблизно рівним однієї секунді. Відразу по закінченню процесу введення коду (по закінченню контрольного проміжку часу) мікропроцесор записує прийнятий у такий спосіб код в EEPROM. Код являє собою послідовність байтів, що відображають усі стани клавіатури під час набору. Після того, як коди будуть записані, замок можна перевести в робочий режим. Для цього передбачений спеціальний тумблер вибору режимів.

У робочому режимі замок чекає введення коду. Для відкривання дверей необхідно повторити ті ж самі маніпуляції із кнопками, які ви робили в режимі запису. Мікроконтролер так само, як і в попередньому випадку, відслідковує ці маніпуляції й записує отриманий у такий спосіб код в ОЗП. По закінченню введення коду (після закінчення контрольного проміжку часу) програма переходить у режим звірення коду, що перебуває в ОЗП, і коду, записаного в EEPROM. Спочатку порівнюється довжина обох кодів. Потім коди звіряються побайтно. Якщо порівняння пройшло успішно, мікроконтролер подає сигнал на механізм відкривання замка.

Отже, сформулюємо завдання в такий спосіб:

«Створити схему й програму електронного кодового замка, що має десять кнопок для введення коду, позначених цифрами від «0» до «9». Замок повинен мати перемикач режимів «Запис/Робота». У випадку правильного набору коду замок повинен включати виконавчий механізм замка (соленоїд або електромагнітну засувку). Введення коду повинно проводитися описаним вище способом».

Алгоритм

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 48</i>

При складанні даного алгоритму нам не обійтися без такого поняття, як **«код стану клавіатури»**. Що таке код стану?. Усі кнопки клавіатури підключаються до мікроконтролера за допомогою портів вводу-виводу. Для підключення десяти кнопок (кнопки «0» - «9») одного порту недостатньо. Кілька кнопок доведеться підключити до другого.

Контролер читає вміст цих портів і одержує код, що відповідає їхньому стану. Кожній кнопці клавіатури в цьому коді буде відповідати свій окремий біт. Коли кнопка натиснута, відповідний біт буде дорівнює нулю. Коли відпущена – одиниці. Тому при різних комбінаціях натиснутих і відпущених кнопок код стану клавіатури буде мати різні значення.

У момент включення живлення всі кнопки замка повинні бути відпущені. Якщо це не так, то виникає невизначеність у роботі замка. Тому наш **алгоритм повинен починатися** із процедури очікування відпускання всіх кнопок. Як тільки всі кнопки виявляться відпущеними або у випадку, якщо вони взагалі не були натиснуті, починається інша **процедура очікування**. Цього разу програма очікує моменту натискання кнопок. Це саме той режим роботи, у якому замок буде перебувати більшу частину часу. У момент натискання кожної із кнопок починається цикл уведення ключової комбінації.

Процедура введення ключової комбінації являє собою багаторазово повторюваний процес, що періодично зчитує код стану клавіатури. Щораз після чергового зчитування коду програма перевіряє, чи не змінився цей код. Як тільки код зміниться, нове його значення записується в чергову комірку ОЗП. У результаті, поки стан клавіатури не змінюється, програма перебуває в режимі очікування.

Як тільки стан змінився, відбувається запис нового значення коду стану в пам'ять. Тому ключова комбінація, записана в ОЗП, буде являти собою перерахування всіх значень коду стану клавіатури, який

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 49</i>

він ухвалював у процесі введення ключової комбінації. Тривалість же втримання кнопок у кожному зі станів у пам'ять не записується.

Однак це лише приблизний алгоритм процедури введення ключової комбінації. Так сказати, її кістяк. Насправді алгоритм небагато складніше. Виявивши зміну стану клавіатури, програма не відразу записує новий код в ОЗП. З метою боротьби з брязком контактів, а також для компенсації неточності одночасного натискання декількох кнопок, програма спочатку витримує спеціальну захисну паузу, потім повторно зчитує код стану клавіатури й лише після цього записує новий код в ОЗП.

Тривалість захисної паузи обрана рівної 48 мс. Така пауза особливо корисна у випадку, якщо при наборі ключової комбінації ви прагнете використовувати одночасне натискання кнопок. Як би ви не намагалися натиснути кнопки одночасно, вам цього не вдасться. Однаково буде якась розбіжність у моменті замикавання контактів. Причому порядок замикавання контактів буде залежати від багатьох факторів і практично є випадковим.

Якщо не прийняти спеціальних заходів, то в момент такого натискання програма зафіксує не одне, а кілька послідовних змін коду стану клавіатури. Якщо отримана в такий спосіб кодова комбінація буде записана в EEPROM, то відкрити такий замок буде практично неможливо.

При спробі повторити ті ж натискання, замикавання контактів будуть відбуватися в іншому порядку. Програма сприйме його як зовсім інший код. Захисна пауза вирішує цю проблему. У якому б порядку не замикалися контакти при одночаснім натисканні декількох кнопок, після паузи всі ці процеси закінчуються. Повторне зчитування дасть уже усталений код стану клавіатури. Повторити таку комбінацію не важко буде.

Крім захисної паузи, для боротьби із брязком застосовується **багаторазове зчитування коду стану**. Тобто насправді щораз

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 50</i>

відбувається не одне, а кілька послідовних операцій по зчитуванню коду стану. Зчитування відбувається доти, поки кілька раз підряд буде отриманий той самий код.

Тепер поговоримо про те, **як програма виходить із процедури введення ключової комбінації**. Як уже говорилося раніше, для виходу з процедури використовується захисний проміжок часу. Для формування цього проміжку застосовується **таймер**. Таймер повинен працювати в режимі Normal. У цьому режимі він просто рахує тактові імпульси.

Процедура введення кодової комбінації влаштована таким чином, що при кожному натисканні або відпусканні кожної із кнопок таймер скидається в нуль. У проміжку між натисканнями його показання збільшуються під дією тактового сигналу. Якщо протягом захисного проміжку часу не буде натиснута жодна кнопка, показання таймера збільшаться до контрольної межі. Програма постійно перевіряє цю умову. Як тільки показання лічильника перевищать контрольну межу, процедура введення кодової комбінації завершується. Величина контрольного проміжку часу дорівнює 1 с.

Подальші дії після виходу із процедури введення кодової комбінації визначаються станом перемикача режимів роботи. Якщо контакти перемикача замкнені, програма переходить до процедури запису кодової комбінації в EEPROM.

Спочатку в EEPROM записується довжина кодової комбінації. А потім байт за байтом і сама комбінація. Якщо контакти перемикача режиму роботи розімкнуті, то програма переходить до процедури перевірки коду. Ця процедура спочатку дістає з EEPROM записану раніше довжину кодової комбінації й порівнює її з довжиною тільки що введеної комбінації.

Якщо дві ці величини не рівні, процедура перевірки коду відразу ж завершується з негативним результатом. Якщо довжина обох комбінацій однакова, то програма приступає до побайтному їх

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 51</i>

порівнянню. Для цього вона по черзі зчитує з EEPROM раніше записані туди байти й порівнює кожний з них з відповідними байтами в ОЗП. При першій же розбіжності процес порівняння також завершується. І завершується негативно.

І тільки в тому випадку, якщо всі байти в ОЗП й в EEPROM виявляться однаковими, порівняння вважається успішним. У випадку успішного порівняння програма переходить до **процедури відкривання замка**. Процедура відкривання починається з видачі відкриваючого сигналу на виконавчий механізм. Потім програма витримує паузу в 2 с і знімає сигнал. Цього часу досить для того, щоб відкрити двері. Потім замок переходить у вихідний стан.

Схема

Можливий варіант схеми замка наведений на **рис. 5.1**. Кнопки S1-S10 служать для набору коду. Перемикач S11 призначений для вибору режиму роботи. Якщо контакт перемикача S11 замкнути, замок переходить у режим «Запис». Розімкнуті контакти відповідають режиму «Робота».

Схема керування механізмом замка складається із транзисторного ключа VT1 і електромагнітного реле K1. Резистор R1 обмежує струм бази ключа. Діод VD1 служить для захисту від напруги самоіндукції, що виникає на котушці реле. Живлення реле здійснюється від окремого джерела +12 В (живлення мікроконтролера +5 В). Якщо в якості VT1 застосовувати транзистор КТ315, то електромагнітне реле може мати робочу напругу +12В і робочий струм не більше 250 мА. Контакти реле повинні бути розраховані на керування виконавчим механізмом (соленоїдом).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 52

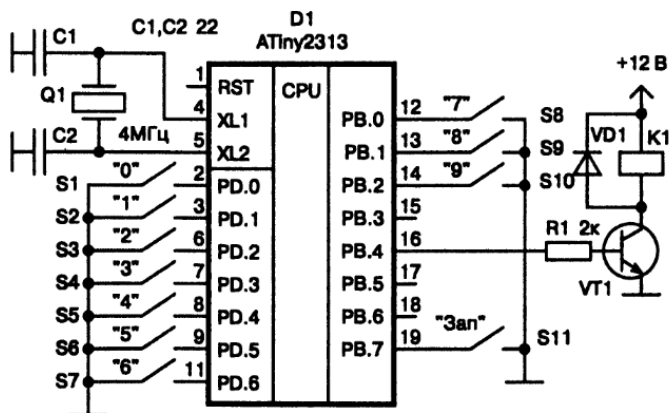


Рис. 5.1. Схема кодового замка

Обратите внимание, что в данной схеме одни линии порта PB будут работать как входы, а другие (в частности линия PB.4) – как выходы. При распределении выводов порта между периферийными устройствами учитывалась возможность объединения замка с музыкальной шкатулкой. В этом случае шкатулка может управляться одной кнопкой и служить дверным звонком.

Программа на Ассемблере

Один из возможных вариантов программы на Ассемблере приведен в листинге 1.19. Прежде чем переходить к подробному описанию ее работы, разберемся в некоторых общих вопросах. Начнем с **кода состояния клавиатуры**. Как уже говорилось ранее, код состояния должен иметь отдельный бит для каждой кнопки клавиатуры. Итого, получается десять битов.

Одного байта явно мало. Значит, мы должны использовать двухбайтовый код состояния. Самый простой способ получить двухбайтовый код состояния – это прочитать сначала содержимое

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 53</i>

порта РО, а затем – содержимое порта РВ. Затем нужно наложить на каждый из полученных байтов маску.

Маска должна обнулить все ненужные нам разряды и оставить разряды, к которым подключены наши кнопки. Для числа, прочитанного из порта РЭ, маска должна обнулить самый старший разряд и оставить все остальные. Для числа, прочитанного из порта РВ, нужно, напротив, оставить три младших разряда и обнулить все остальные. Полученные таким образом два байта мы и будем считать кодом состояния клавиатуры.

В том случае, если все десять кнопок (S1-S10) отпущены, код состояния клавиатуры равен 0x7F, 0x07 (0b01111111, 0b00000111). В таком коде значащие биты, отражающие состояние той или иной кнопки, равны единице, а все остальные биты равны нулю. Если нажать любую кнопку, то код состояния изменится. Соответствующий этой кнопке бит примет нулевое значение. Таким образом, любое изменение состояния клавиатуры вызовет соответствующее изменение кода.

Теперь вернёмся к тексту программы. В программе применяются следующие новые для нас операторы.

cli – Общий запрет прерываний. Действие данной команды обратно действию уже знакомой нам команды sei. Команда не имеет параметров и служит для сброса флага I в регистре SREG.

St – Косвенная запись в память. Команда имеет три модификации: st W,Rd; st W+,Rd; st -W,Rd, где W – это одна из регистровых пар (X, Y или Z). Rd – имя одного из регистров общего назначения. Независимо от модификации команда выполняет запись содержимого регистра Rs в ОЗП по адресу, который хранится в регистровой паре W.

При этом первая модификация команды не изменяет содержимое регистровой пары W. Вторая модификация увеличивает содержимое регистровой пары на единицу после того, как произойдет запись. А третья модификация команды уменьшает на единицу

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 54</i>

содержимое регистровой пары перед тем, как произойдет запись в ОЗП.

В строке 75 нашей программы (см. листинг 1.19) содержимое регистра ХЬ записывается в ОЗУ по адресу, который хранится в регистровой паре Ъ. После этого содержимое регистровой пары Ъ увеличивается на единицу.

Id – Косвенное чтение из памяти. Данная операция является обратной по отношению к предыдущей. Она тоже имеет три модификации:

Id Rd,W; Id Rd, W+; Id Rd,-W

Операция производит чтение байта из ячейки ОЗУ, адрес которой хранится в регистровой паре W (то есть X, Y или Z) и записывает прочитанный байт в регистр общего назначения Rd. Содержимое регистровой пары так же, как и в предыдущем случае, ведет себя по-разному, в зависимости от модификации команды. То есть оно либо не изменяется, либо увеличивается после чтения, либо уменьшается прежде, чем байт будет прочитан.

В строке 97 нашей программы (листинг 1.19) читается байт из ячейки ОЗУ, адрес которой хранится в регистровой паре Z, и записывается в регистр data. Затем содержимое регистровой пары Z увеличивается на единицу.

Brsh – переход по условию «больше или равно». В качестве условия для перехода выступает содержимое флага переноса C. Флаг переноса устанавливается по результатам операции сравнения или вычитания. Команда имеет всего один параметр – относительный адрес перехода. Переход выполняет в том случае, если флаг переноса равен нулю. А это происходит только тогда, когда в предшествующей операции сравнения (вычитания) второй операнд окажется больше или равен первому.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 55</i>

Sbic – оператор типа «проверить - пропустить». Общая форма записи команды: **sbic** А,п, где А - номер регистра ввода-вывода; п - номер разряда.

Вместо номера регистра и номера разряда может использоваться имя регистра и имя разряда. Обычно используются стандартные имена от фирмы Atmel. Команда проверяет содержимое разряда номер п регистра А. Если разряд сброшен, то очередная команда программы не выполняется.

Пример использования данной команды - строка 159 нашей программы (см. листинг 1.19). В этой строке команда **sbic** проверяет бит *eewe* регистра *eesc*. Если этот бит сброшен, то команда в строке 160 не выполняется, а управление передается к строке 161. Если бит установлен, то выполняется команда в строке 160. Команда **sbic** имеет одно ограничение. Она работает с регистрами ввода-вывода с адресами в диапазоне от 0 до 31.

Cbr – сброс разрядов РОН. Данный оператор предназначен для одновременного сброса нескольких разрядов. Оператор имеет два параметра. Первый параметр - это имя регистра общего назначения, разряды которого должны быть сброшены. Второй параметр - это маска сброса разрядов. В данном случае маска – это двоичное число, у которого в единицу установлены те разряды, которые должны быть сброшены. Например, в строке 129 программы (листинг 1.19) сбрасываются разряды регистра ХН. Значение маски равно 0xF8. В двоичном виде число 0xF8 выглядит так: 0b11111000. Поэтому в результате действия команды **cbr** в строке 129 пять старших разрядов числа, находящегося в регистре ХН, будут сброшены в ноль, а три младшие останутся без изменений. Это альтернативный способ наложения маски.

Листинг 5.1.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 56

```

: #####
: ##          Пример 10          ##
: ##          Кодовый замок      ##
: #####

, ----- Псевдокоманды управления

1 .include "tn2313def inc"      ; Присоединение файла описаний
2 list                          ; Включение листинга

; ----- Модуль описаний

3 .def      drebL = R1          ; Буфер антидребезга младший байт
4 .def      drebH = R2          ; Буфер антидребезга старший байт
5 .def      temp = R16          ; Вспомогательный регистр
6 .def      data = R17          ; Регистр передачи данных
7 .def      flz = R18           ; Флаг задержки
8 .def      count = R19         ; Регистр передачи данных
9 .def      addre = R20         ; Указатель адреса в EEPROM
10 .def     codL = R21           ; Временный буфер кода младший байт
11 .def     codH = R22           ; Временный буфер кода старший байт

; ----- Определение констант

12 .equ     bsize = 60           ; Размер буфера для хранения кода
13 .equ     kzad = 3000         ; Константа, определяющая длительность защитной паузы
14 .equ     kandr = 20          ; Константа антидребезга

; ----- Резервирование ячеек памяти (SRAM)

15         .dseg                ; Выбираем сегмент ОЗУ
16         .org      0x60         ; Устанавливаем текущий адрес сегмента

17 bufr:   .byte    bsize        ; Буфер для приема кода

; ----- Резервирование ячеек памяти (EEPROM)

```


Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 57</i>

```

18      .eseg      ; Вибираем сегмент EEPROM
19      .org      0x08      ; Устанавливаем текущий адрес сегмента

20      klen:    .byte    1      ; Ячейка для хранения длины кода
21      bufe:    .byte    bsize   ; Буфер для хранения кода

;----- Начало программного кода

22      .cseg
23      org      0      ; Выбор сегмента программного кода
; Установка текущего адреса на ноль

24      start:   rjmp     init    ; Переход на начало программы
25      reti     ; Внешнее прерывание 0
26      reti     ; Внешнее прерывание 1
27      reti     ; Таймер/счетчик 1, захват
28      rjmp     propr    ; Таймер/счетчик 1, совпадение, канал А
29      rjmp     propr    ; Таймер/счетчик 1, прерывание по переполнению
30      reti     ; Таймер/счетчик 0, прерывание по переполнению
31      reti     ; Прерывание UART прием завершен
32      reti     ; Прерывание UART регистр данных пуст
33      reti     ; Прерывание UART передача завершенна
34      reti     ; Прерывание по компаратору
35      reti     ; Прерывание по изменению на любом контакте
36      reti     ; Таймер/счетчик 1 Совпадение, канал В
37      reti     ; Таймер/счетчик 0. Совпадение, канал В
38      reti     ; Таймер/счетчик 0. Совпадение, канал А
39      reti     ; USI готовность к старту

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 58</i>

```

40         reti          ; USI Переполнение
41         reti          ; EEPROM Готовность
42         reti          ; Переполнение охранный таймера

;-----
; *
; *          Модуль инициализации
; *
; *
;-----

43  init:
;----- Инициализация стека
44         ldi          temp, RAMEND ; Выбор адреса вершины стека
45         out          SPL, temp    ; Запись его в регистр стека

;----- Инициализация портов В/В
46         ldi          temp, 0x18   ; Инициализация порта PB
47         out          DDRB, temp
48         ldi          temp, 0xE7
49         out          PORTB, temp

50         ldi          temp, 0x7F   ; Инициализация порта PD
51         out          PORTD, temp
52         ldi          temp, 0
53         out          DDRD, temp

;----- Инициализация (выключение) компаратора
54         ldi          temp, 0x80
55         out          ACSR, temp

;----- Инициализация таймера

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 59</i>

```

56      ldi    temp, high(kzad )      ; Записуємо коефіцієнт затримки
57      out    OCR1AH, temp
58      ldi    temp, low(kzad )
59      out    OCR1AL, temp
60      ldi    temp, 0x03             ; Вибір режиму роботи таймера
61      out    TCCR1B, temp

```

```

;*****
; *
; *          Начало основной программы
; *
;*****

```

```

62 main:  ldi    codL, 0x7F           ; Код для порівняння (молодший байт)
63        ldi    codH, 0x07           ; Код для порівняння (старший байт)

64 m0:    rcall  incod                ; Введення та перевірка коду клавіш
65        brne  m0                    ; Якщо хоча б одна не натиснута, продовжуємо

66 m1:    rcall  incod                ; Введення та перевірка коду клавіш
67        breq  m1                    ; Якщо не одна не натиснута, продовжуємо

68 m2:    ldi    ZH, high(bufR)       ; Встановлення вказівника на початок буфера
69        ldi    ZL, low(bufR)
70        clr    count                ; Скидання лічильника байт

```

```

;----- Цикл введення коду

```

```

71 m3:    cli    data, 1              ; Заборона всіх переривань
72        ldi    data, 1              ; Викликаємо затримку першого типу
73        rcall  wait                 ; К підпрограму затримки

74 m5:    rcall  incod                ; Введення та перевірка коду кнопок
75        st    Z+, XL                ; Записуємо його в буфер
76        st    Z+, XH
77        inc   count                 ; Збільшення лічильника байт
78        inc   count
79        cpi   count, bsize          ; Перевіряємо, чи кінець буфера
80        brsh  m7                    ; Якщо кінець, завершуємо введення коду

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 60</i>

```

81      mov    codL, XL      ; Записуємо код як старий
82      mov    codH, XH

83      ldi    data, 2      ; Виводимо задержку третього типу
84      rcall  wait        ; К підпрограмі задержки

85  m6:   rcall  incod      ; Ввод і перевірка коду кнопок
86      brne  m3          ; Якщо змінилося, записуємо в буфер
87      cpi   flz, 1      ; Перевірка закінчення фази вводу коду
88      brne  m6

; ----- Опрос состояния тумблера

89  m7:   sbic  PINB, 7    ; Перевірка стану тумблера
90      rjmp  m9          ; К процедурі перевірки коду

; ----- Процедура запису коду

91      mov    data, count ; Поміщаємо довжину коду в data
92      ldi    addre, klen ; Адрес в EEPROM для збереження довжини коду
93      rcall  eewr       ; Записуємо в довжину коду EEPROM

94      ldi    addre, bufe ; В регістр адреси початок буфера в EEPROM
95      ldi    ZH, high(buf) ; В регістрову пару Z записуємо
96      ldi    ZL, low(buf) ; адрес початку буфера в ОЗУ

97  m8:   ld    data, Z+   ; Читаємо наступний байт з ОЗУ
98      rcall  eewr       ; Записуємо в довжину коду EEPROM
99      dec    count      ; Декремент лічильника байтів
100     brne  m8         ; Якщо не кінець, продовжуємо запис

101     rjmp  m11        ; К процедурі відкриття замка

; ----- Процедура перевірки коду

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 61</i>

```

102 m9:   ldi   addr, klen      ; Адрес хранения длины кода
103       rcall  eerd        ; Чтение длины кода из EEPROM
104       cp     count, data  ; Сравнение с новым значением
105       brne  m13         ; Если не равны, к началу

106       ldi   addr, bufe   ; В YL начало буфера в EEPROM
107       ldi   ZH, high(buf) ; В регистровую пару Z записываем
108       ldi   ZL, low(buf)  ; адрес начала буфера в ОЗУ

109 m10:  rcall  eerd        ; Читаем очередной байт из EEPROM
110       ld    temp, Z+      ; Читаем очередной байт из ОЗУ

111       cp     data, temp   ; Сравняем два байта разных кодов
112       brne  m13         ; Если не равны, переходим к началу

113       dec   count        ; Уменьшаем содержимое счетчика байтов
114       brne  m10         ; Если не конец, продолжаем проверку

; ----- Процедура открывания замка

115 m11:  sbi   PORTB, 4     ; Команда "Открыть замок"
116       ldi   data, 3      ; Вызываем задержку третьего типа
117       rcall  wait
118       cbi   PORTB, 4     ; Команда "Закреть замок"

119 m13:  rjmp  main        ; Перейти к началу

; -----
; *
; *      Вспомогательные процедуры      *
; *
; -----
; ----- Ввод и проверка 2 байтов с клавиатуры

120 incod: push   count
121       ldi   XL, 0        ; Обнуление регистровой пары X

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 62</i>

```

122         ldi     XH, 0
123 ic1:     ldi     count, kandr ; Константа антидребезга
124         mov     drebl, XL      ; Старое значение младший байт
125         mov     drebh, XH      ; Старое значение старший байт

126 ic2:     in     XL, PIND      ; Вводим код (младший байт)
127         cbr    XL, 0x80      ; Накладываем маску
128         in     XH, PINB      ; Вводим код (старший байт)
129         cbr    XH, 0xF8      ; Накладываем маску

130 ic3:     cp     XL, drebl     ; Сверяем младший байт
131         brne   ic1           ; Если не равен, начинаем с начала
132         cp     XH, drebh     ; Сверяем старший байт
133         brne   ic1           ; Если не равен, начинаем с начала

134 ic4:     dec    count        ; Уменьшаем счетчик антидребезга
135         brne   ic2          ; Если еще не конец, продолжаем

136         cp     XL, codL      ; Сравнение с временным буфером
137         brne   ic5          ; Если не равно, заканчиваем сравнение
138         cp     XL, codH      ; Сравниваем старшие байты

139 ic5:     pop    count
140         ret

; ----- Подпрограмма задержки

141 wait:    cpi     data, 1      ; Проверяем код задержки
142         brne   w1
143         ldi    temp, 0x40     ; Разрешаем прерывание по совпадению
144         rjmp   w2
145 w1:      ldi    temp, 0x80     ; Разрешаем прерывания по переполнению

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 63

```

146 w2:    out    TIMSK, temp    ; Записуємо маску
147        clr    temp
148        out    TCNT1H, temp ; Обнуляємо таймер
149        out    TCNT1L, temp

150        ldi    flz, 0        ; Сбрасуємо флаг задержки
151        sei    ; Разрешаємо прерывания
152        cpi    data, 2       ; Если это задержка 2-го типа
153        breq   w4            ; Переходим к концу подпрограммы

154 w3:    cpi    flz, 1        ; Ожидание окончания задержки
155        brne   w3

156        cli    ; Запрещаем прерывания

157 w4:    ret    ; Завершаем подпрограмму

; ----- Запись байта в ячейку EEPROM

158 eewr:   cli    ; Запрещаем прерывания
159        sbic   EECR, EWE     ; Проверяем готовность EEPROM
160        rjmp  eewr          ; Если не готов ждем
161        out    EEAR, addr    ; Записываем адрес в регистр адреса
162        out    EEDR, data    ; Записываем данные в регистр данных
163        sbi    EECR, EEMWE   ; Устанавливаем бит разрешения записи
164        sbi    EECR, EWE     ; Устанавливаем бит записи
165        inc   addr          ; Увеличиваем адрес в EEPROM
166        ret    ; Выход из подпрограммы

; ----- Чтение байта из ячейки EEPROM

167 eerd:   cli    ; Запрещаем прерывания
168        sbic   EECR, EWE     ; Проверяем готовность EEPROM
169        rjmp  eerd          ; Если не готов ждем

170        out    EEAR, addr    ; Устанавливаем бит инициализации чтения
171        sbi    EECR, EERE    ; Устанавливаем бит инициализации чтения
172        in    data, EEDR     ; Помещаем прочитанный байт в data
173        inc   addr          ; Увеличиваем адрес в EEPROM
174        ret    ; Выход из подпрограммы

; *****
; *
; *      Процедура обработки прерываний      *
; *
; *
; *****

; ----- Прерывание по совпадению

175 propr: ldi    flz, 1        ; Установка флага задержки
176        reti   ; Завершаем обработку прерывания

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 64</i>

Опис програми (лістинг 5.1)

Рядки 1, 2 програми, я думаю, питань не викликають. У **рядках 3–11** відбувається опис усіх використовуваних у програмі змінних. Призначення кожної із цих змінних ми розглянемо в ході опису принципів роботи програми. Далі, у **рядках 12–14** відбувається опис констант. Кожна із трьох констант визначає один з параметрів нашого пристрою. Зупинимося на цьому докладніше.

Константа bsize (рядок 12) визначає **розмір буфера для зберігання кодової комбінації**. Цей розмір обраний рівним 60 гніздам. Враховуючи, що в буфер будуть записуватися коди стану клавіатури, а кожний такий код складається із двох байтів, у буфер зазначеного розміру можна записати послідовність із 30 кодів.

Якщо враховувати, що запису підлягає кожна зміна коду стану, а його зміна відбувається як при натисканні кнопки, так і при її відпусканні, то після натискання й відпускання однієї із кнопок у буфер запишеться два коди.

Виходить, обсягу буфера нам вистачить на 15 послідовних натискань. Цього цілком достатньо, тому що типова кодова комбінація полягає звичайно з 4–5 цифр. Якщо ви вважаєте, що цього недостатньо, ви можете збільшити розмір буфера, просто помінявши значення константи **bsize** в **рядку 12**. Максимально можливий розмір обмежений обсягом ОЗУ й рівний приблизно 100 байтам (повний розмір ОЗУ 128 байт).

Урахуйте, що буфер не може займати весь обсяг ОЗУ, тому що у верхніх адресах необхідно обов'язково залишити простір, який буде використовувати пам'ять. Обертаю вашу увагу, що константа **bsize** використовується для завдання розміру не тільки буфера в ОЗУ, але й для завдання розміру буфера в EEPROM, який призначений для довгочасного зберігання кодової комбінації.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 65

Константа kzad (рядок 13) визначає тривалість захисної паузи. Призначення захисної паузи докладно описане вище. Константа являє собою коефіцієнт перерахування для таймера, при яким величина сформованої затримки буде рівна 48 мс.

Константа kandr (рядок 14) – це константа антидрібзгу. Вона використовується в спеціальній антибязковій процедурі. Константа визначає, скільки раз підряд повинен повторитися той самий код стану клавіатури, щоб програма припинила цикл антидрібзга й перейшла до обробки ліченого коду.

Після визначення констант починається **блок резервування оперативної пам'яті (рядка 15–17)**. У **рядку 15** вибирається відповідний сегмент пам'яті, у **рядку 16** устанавлюється показчик на адресу 0x60. Міркування для вибору саме цієї адреси вже приводилися в попередньому прикладі.

Властиво резервування гнізд проводиться в **рядку 17**. Директива `byte` резервує необхідна кількість гнізд ОЗУ, починаючи з адреси, обумовленої міткою `bufg`. У цьому випадку `bufg` буде рівний 0x60. Кількість гнізд визначається константою `bsize`.

Далі, у **рядках 18–21** відбувається резервування гнізд в енергонезалежній пам'яті (EEPROM). У **рядку 18** вибирається сегмент EEPROM. У **рядку 19** устанавлюється поточне значення показчика цього сегмента. Показчику привласнюється значення 0x08. Тобто розміщення даних у пам'яті EEPROM буде починатися з восьмого гнізда. Взагалі в цьому випадку можна було починати з нульового гнізда. Це не зроблене лише з міркувань надійності роботи.

Фірма Atmel не рекомендує без особливої необхідності використовувати гніздо з нульовою адресою, тому що саме вона зазнає найбільшому ризику втрати інформації при неприпустимих перепадах напруги живлення, особливо якщо перепади напруги виникають у момент запису інформації в EEPROM. Тому що

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 66</i>

EEPROM не працює зі стеклом, у нас є запас по гніздах. Тому ми відступили на цілих вісім гнізд.

Властиво команди резервування займають **рядки 20 і 21**. У **рядку 20** резервується одне гніздо, у яким буде зберігатися довжина ключової комбінації. У **рядку 21** резервується буфер довжиною **bsize**, у яким буде зберігатися сама комбінація.

Після резервування гнізд ми переходимо **в сегмент програмного коду (рядок 22)**. І починаємо формування програми з нульової адреси (команда **org** у **рядку 23**). Програмний код починається з **таблиці перевизначення векторів переривань (рядки 24–42)**.

Як бачите, у цьому випадку ми будемо використовувати два види переривань.

Це переривання по збігові в каналі А таймера/лічильника 1 (**рядок 28**) і переривання по переповненню того ж таймера (**рядок 29**). Перше переривання використовується для формування захисної затримки в 48 мс. А друге – для формування контрольного проміжку часу в 1 с. Для обох видів переривань призначена та сама процедура обробки: **prorog**. Чому та сама і як вона працює, ми довідаємося трохи пізніше.

У **рядках 44–61** розташований **модуль ініціалізації**. Починається ініціалізація із програмування портів уведення-виводу (**рядка 46–53**). Усі розряди порту PD і більша частина розрядів порту PV конфігуруються як входи. І лише два розряди PV.3 і PV.4 конфігуруються як виходи.

Розряд PV.4 використовується для керування механізмом замка. А **розряд PV.3** взагалі поки не використовується (зарезервований). Його ми будемо використовувати як вихід звуку, коли будемо поєднувати в один пристрій наш кодовий замок і музичну скриньку. Для всіх розрядів обох портів, настроєних на

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 67</i>

введення, включаються внутрішні навантажувальні резистори. На обох виходах (РВ.3 і РВ.4) встановлюється низький логічний рівень.

У рядках 54, 55 програмується **аналоговий компаратор**. Програмування таймера/лічильника 1 проводиться в рядках 56–61. Спочатку в обидві половинки регістру збігу (OCR1AH, OCR1AL) записується код, що визначає тривалість захисної затримки (рядка 56–59). Потім у регістр стану TCCR1B записується код 0x03 (рядка 60, 61). При записі цього коду таймер/лічильник 1 переводиться в режим Normal з використанням попереднього дільника. А коефіцієнт попереднього розподілу стає рівним 1/64.

Рядки 62–119 займає **основний цикл програми**, рядка 120–174 – **набір допоміжних підпрограм**, а рядка 175 і 176 – **процедура обробки переривань**. Розгляд програми зручніше почати з допоміжних процедур.

І перша процедура, яку ми розглянемо – це **процедура введення й попередньої обробки коду стану клавіатури**. Процедура являє собою підпрограму з іменем **incod** і розташовано в рядках 120–140. У процесі роботи процедура використовує регістрову пару X і дві допоміжні регістрові пари: **codh, codl** (описано в рядках 10, 11) і **drebh, drebl** (описано в рядках 3, 4).

Головне завдання даної процедури – одержати **код стану клавіатури**, використовуючи алгоритм багаторазового зчитування для боротьби з антидрибзгом. Крім того, процедура виконує ще одну, допоміжну функцію. Вона порівнює отриманий описаним вище способом код стану клавіатури з іншим кодом, який зберігається в парі регістрів **codh-codl**. Таке порівняння використовується в основній програмі для оцінки значення коду стану.

Таким чином, процедура **incod** по закінченню своєї роботи повертає два різні значення. По-перше, код стану клавіатури (у регістровій парі X), а по-друге, результат порівняння коду з

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 68</i>

контрольним значенням (використовуючи прапор нульового результату Z).

Розглянемо роботу процедури докладніше. По-перше, підпрограма використовує традиційне збереження й відновлення вмісту реєстрів у стеці. Цього разу збереженню підлягає всього один реєстр – реєстр **count**. У рядку 120 його значення зберігається, а в рядку 139 – відновлюється. Основний же текст підпрограми складається із двох частин:

- у першій частині (рядка 121–133) реалізується алгоритм уведення коду й боротьби з антибрязком;
- у другій частині (рядка 134–138) проводиться порівняння отриманого коду із числом у буфері `codh-codl`.

Почнемо із процедури введення коду й боротьби з антидрибізгом. Ця процедура являє собою нескінченний цикл, який при кожному проході робить формування коду стану клавіатури й при цьому підраховує, скільки раз підряд отриманий код буде мати однакове значення. Для підрахунку використовується реєстр **count**. Причому на початку в **count** записується константа **kandr**, і щораз, коли новий код рівний попередньому, вміст **count** зменшується на одиницю (використовується зворотний рахунок).

Якщо при черговому проході код стану змінить своє значення, то в реєстр **count** знову записується константа **kandr**, і підрахунок починається спочатку. Закінчується цикл зчитування кодів лише тоді, коли значення **count** досягнеться нуля. Нове значення коду стану клавіатури формується в реєстровій парі X. Для того, щоб нове значення можна було порівнювати зі старим, старе записується в буфер `drebh-drebl`.

Тепер текст описаної вище програми розглянемо по черзі.

Усе починається з підготовки реєстрової пари X до прийняття нового значення коду. У рядках 121, 122 вміст X обнуляється.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 69</i>

Нескінченний цикл багаторазового зчитування займає **рядки 123–135**.

У **рядку 123** у реєстр count записується початкове значення kandr. У **рядках 124, 125** уміст реєстрової пари X зберігається в буфері drebh-drebl. У такий спосіб запам'ятовується старе значення коду стану перед тем, як буде отримано нове. У **рядках 126–129** відбувається зчитування кодів з портів **PD** і **PB** і накладення масок. Отриманий у результаті цих операцій код стану клавіатури виявиться в реєстровій парі X.

У **рядках 130–133** проводиться порівняння старого й нового значень кодів. Порівняння відбувається побайтно. Спочатку рівняються молодші байти (**рядка 130,131**), потім старші (**рядка 132, 133**). Оператор bne виконує перехід за умовою «не рівно». Тому, якщо хоча б одна з операцій порівняння дасть позитивний результат (коди виявляться неоднаковими), те керування перейде по мітці ісі. Тобто до того місця програми, де лічильнику count знову привласнюється значення kandr.

Якщо ж обидві операції порівняння дадуть негативні результати коди виявляться рівними, керування перейде до **рядка 134**. У **рядку 134** відбувається зменшення вмісту реєстру count на одиницю. Після цього проводиться перевірка на нуль (**рядок 135**). Якщо після зменшення вмісту count воно ще не дорівнює нулю, то оператор bne у **рядку 135** передає керування по мітці ісі2, і цикл антидрибізгу триває. А якщо ні, то цикл завершиться, і керування переходить до **рядка 136**.

У **рядках 136–138** перебуває процедура порівняння тільки що отриманого значення коду стану клавіатури із числом у буфері codh-codl. Порівняння проходить побайтно. Спочатку в **рядку 136** рівняються молодші байти. Якщо вони не рівні, то подальше порівняння не має змісту. Тому керування передається по мітці ісі5, і підпрограма завершується.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 70</i>

Якщо молодші байти порівнюваних величин виявилися рівні, то остаточний результат порівняння тепер можна одержати, просто зрівнявши між собою старші байти. Це порівняння проводиться в **рядку 138**.

У результаті, при виході з підпрограми incod:

- прапор Z буде встановлений, якщо порівнювані коди рівні між собою;
- прапор Z буде скинутий, якщо коди не рівні.

Наступна додаткова процедура, що забезпечує роботу основної частини програми, – це **підпрограма формування затримки**. Для формування тимчасових інтервалів ця процедура використовує таймер. Ми вже розглядали два варіанти подібних процедур. Одна з них використовувала пряме читання вмісту таймера й цикл очікування, а друга використовувала переривання.

У цьому випадку розроблений ще один варіант, який являє собою щось середнє між двома попередніми. У цьому варіанті будуть використовуватися й переривання, і цикл очікування. Новий алгоритм реалізує підпрограма wait, яка займає **рядки 141–157**. Підпрограма має три режими роботи. Номер режиму передається в підпрограму при її виклику.

Для цього він записується в реєстр data. У **режимі номер 1** підпрограма формує затримку 48 мс. У **режимі номер 3** формується затримка в 1 с. У **режимі номер 2** затримка не формується. Підпрограма просто набудовує таймер точно так само, як у режимі номер 3, дозволяє переривання й закінчує свою роботу. Контрольний інтервал часу тривалістю 1 з формується вже поза підпрограмою wait (в основному тексті програми).

Основний **принцип формування затримки** будується на використанні прапора затримки. У якості прапора затримки застосовується реєстр flz. Опис цього реєстру ви можете бачити в **рядку 7** програми. Перед початком циклу затримки в реєстр f lz

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 71</i>

записується нуль. Потім запускається таймер і дозволяється робота одного з видів переривань (переривання по збігові або переривання по переповненню).

У певний момент часу буде викликана **процедура обробки переривання**. Ця процедура (**рядка 156, 176**) запише в реєстр прапора (f lz) одиницю. Одиниця в реєстрі f lz послужить індикатором того факту, що заданий проміжок часу закінчився. Для виявлення моменту закінчення затримки використовується цикл очікування.

У циклі очікування програма постійно перевіряє вміст реєстру f lz. Поки f lz дорівнює нулю, цикл триває. Закінчується цикл у той момент, коли f lz буде дорівнює одиниці.

Для формування різних значень тривалості затримки використовуються різні види переривань. Переривання по збігові використовується для формування затримки в 48 мс. Для цього значення реєстру збігу обраний таким чином, щоб вміст рахункового реєстру досягся цього значення саме через 48 мс. Переривання по переповненню таймера використовується для формування інтервалу часу, рівного однієї секунді. Благо, що при коефіцієнті перерахування попереднього дільника 1/64 і тактовій частоті в 4 МГц переповнення таймера відбудеться саме через 1 с.

Розглянемо **текст підпрограми wait** докладніше. Починається підпрограма з перевірки значення реєстру data (**рядок 141**). Як уже говорилося раніше, за допомогою цього реєстру в підпрограму передається код номера режиму. У цьому випадку потрібно визначити тривалість формованої затримки.

Від цього залежить, який з видів переривань буде активізований. Для активізації того або іншого виду переривань у реєстр маски таймера (TIMSK) ми будемо записувати різні значення маски. У **рядках 141–145** саме й вибирається це значення. Вибір зводиться до перевірки номера режиму.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 72</i>

Якщо вміст data рівно 1, то виконується **рядок 143**, де в якості маски вибирається число 0x40 (переривання по збігові). Обрана маска записується в регістр temp. Оператор безумовного переходу г jmp у **рядку 144** передає керування по мітці w2 для того, щоб «перестрибнути» **рядок 145**, де вибирається інше значення маски.

Якщо код у регістрі data не рівний 1, то керування передається до **рядка 145**, де в регістр temp записується код 0x80. Маска 0x80 дозволяє переривання по переповненню. Докладніше про настроювання таймера дивитися в **главі 1**. У **рядку 146** обране значення маски записується в регістр TIMSK.

У **рядках 147–149** виконується **скидання таймера** (в обидві половини рахункового регістру таймера записується нульове значення). Із цього моменту починається відлік часу затримки. У **рядку 150** скидається в нуль регістр прапора затримки (flz). А в **рядку 151** проводиться глобальний дозвіл переривань. На цьому всі настроювання, необхідні для роботи процедури затримки, закінчуються.

Залишається лише чекати закінчення заданого проміжку часу. Але перш, ніж починати цикл очікування, програма робить ще одну перевірку номера режиму (**рядка 152, 153**). Справа в тому, що для режиму номер 2 цикл очікування організувати не потрібно. У цьому випадку цикл очікування розташовується поза підпрограмою wait, у тілі самої програми. Тому, якщо був заданий режим номер 2, то робота підпрограми в цьому місці повинна закінчуватися.

Оператор spі у **рядку 152** перевіряє вміст регістру data, де зберігається код номера режиму на рівність цифрі 2. Якщо код режиму рівний двом, то оператор умовного переходу в **рядку 153** передає керування на кінець підпрограми. А якщо ні, то програма переходить до циклу очікування.

Цикл очікування займає всього два **рядки (154, 155)**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 73</i>

У рядку **154** проводиться перевірка вмісту регістру прапора (flz) на рівність одиниці. Поки лічильник перебуває в процесі рахунку й переривання ще не спрацювало, то вміст регістру flz дорівнює нулю, оператор у **рядку 155** передає керування назад на **рядок 154**, і цикл триває. Як тільки процедура обробки переривання запише в flz одиницю, цикл завершується, і керування переходить до **рядка 156**. У цьому **рядку** відбувається глобальна заборона всіх переривань. А в **рядку 157** підпрограма **wait** завершується.

Дві, що залишилися, ще не описані допоміжні процедури призначені для роботи з EEPROM. При записі в цей вид пам'яті й читання з неї потрібно дотримувати певної послідовності дій. Ця послідовність докладно описана в документації на мікроконтролери AVR (див. главу 1).

Там же наведені приклади процедур, рекомендовані виробником для цих цілей. Описувані нижче процедури є практично повною копією рекомендованих процедур, доповнені лише командою автоматичного збільшення адреси. Для керування пам'яттю EEPROM використовуються спеціальні регістри введення-виводу:

EEAR – регістр адреси; EEDR – регістр даних; EECR – регістр керування. Окремі розряди регістру керування також мають свої власні імена: EEWE – біт запису;

EEMWE – біт дозволу записи; EERE – біт читання. Усі назви введені фірмою-виробником і правильно розуміються транслятором, якщо ви не забули приєднати на початку програми файл описів.

Порядок запису байта в EEPROM наступний. Байт даних, призначений для запису, повинен бути поміщений у регістр EEDR, а байт адреси – у регістр в EEAR. Для того, щоб дозволити запис, необхідно встановити біт EEMWE. Потім протягом чотирьох машинних циклів (тобто наступної ж командою) потрібно встановити біт EEWE. Відразу ж після установки біта EEWE починається процес запису. Цей процес займає досить тривалий час.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 74</i>

Усе це час біт EEWЕ залишається встановленим. По закінченню процесу записи він сам скидається в нуль. Такий багатоступінчастий алгоритм придуманий для запобігання випадкового запису.

Підпрограма, що реалізує описаний вище алгоритм запису байта, називається `ewwг` і займає **рядки 158–166**. Байт даних, призначений для запису, передається в процедуру за допомогою регістру `data`, а адреса гнізда, куди потрібно записати дані, – через регістр `addre`. Робота підпрограми починається із глобальної заборони всіх переривань (**рядок 158**).

Це обов'язкова умова роботи з EEPROM. Невчасно викликане переривання може перешкодити процесу запису. У цьому випадку заборона переривань є надлишковим заходом, тому що програма побудована таким чином, що при записі в EEPROM переривання завжди заборонені.

У **рядках 159,160** розташований цикл перевірки готовності EEPROM. Якщо біт EEWЕ установлений, це значить, що попередня операція запису ще не кінчена. Тому в **рядку 159** перевіряється значення цього біта. Поки значення біта дорівнює одиниці, виконується команда безумовного переходу в **рядку 160**, і перевірка виконується знову й знову.

Коли значення біта виявиться рівним нулю, **рядок 160** буде пропущена (спрацює команда `sbic` у **рядку 159**), а цикл очікування перерветься. У **рядку 161** відбувається запис адреси з регістру `addre` у регістр EEAR. У **рядку 162** у регістр EEDR записується байт даних з регістру `data`.

У **рядку 163** установлюється біт дозволу записи. У **рядку 164** – біт записи. Після установки цього біта процес записи буде запущений. Запис буде йти своєю чергою, а програма може продовжувати свою роботу. Головне – не міняти вміст регістрів EEDR і EEAR, поки процес записи не закінчиться. У **рядку 165** відбувається збільшення вмісту регістру `addre`.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 75</i>

І, нарешті, у **рядку 166** підпрограма завершується. Команда в **рядку 165** не ставиться до алгоритму запису в EEPROM. Але її застосування дозволяє використовувати підпрограму `eewt` для послідовного запису ланцюжка байтів, у чому ми й переконаємося далі.

Порядок читання байта набагато простіше. Досить у реєстр `EEAR` записати адреса гнізда, вміст якої потрібно прочитати, а потім установити біт читання (`EERE`). Прочитаний байт автоматично міститься в реєстр `EEDR`.

Підпрограма читання байта з EEPROM називається `eerd` і займає **рядки 167–174**. Адреса гнізда, призначеної для читання, передається в підпрограму через реєстр `addr`. Прочитаний байт даних підпрограма повертає в реєстрі `data`. Починається підпрограма читання, як і підпрограма запису із заборони переривань (**рядок 167**).

Тому що читанню може передувати запис, перш ніж змінювати значення реєстру `EEAR`, потрібно перевірити, чи закінчився процес запису. Тому в **рядках 168, 169** ми бачимо вже знайомий нам цикл очікування готовності `EEPROM`. У **рядку 170** у реєстр `EEAR` записується вміст реєстру `addr`.

У **рядку 171** установлюється біт читання (`EERE`). Як тільки цей біт буде встановлений, моментально відбувається процес читання, і прочитаний байт даних з'являється в реєстрі `EEDR`. У **рядку 172** цей байт міститься в реєстр `data`. У **рядку 173** відбувається збільшення реєстру адреси `addr`. Зміст цього збільшення такий же, як і в попередньому випадку. Тільки тепер подібне приймання дозволяє читати з `EEPROM` ланцюжок байтів. У **рядку 174** підпрограма `eewt` завершується.

Тепер **перейдемо до основної частини програми**. Як уже говорилося, вона займає **рядки 62–119**. І перше, що виконує основна програма, – **цикл очікування відпускання кнопок**. У циклі використовується описана вище підпрограма `incod`. Вона буде не

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 76</i>

тільки зчитувати код стану клавіатури, але й відразу ж робити його порівняння.

Якщо ви не забули, код стану клавіатури при повністю відпущених кнопках рівний 0x7F, 0x07. У рядках 62, 63 цей код записується в допоміжний буфер `codl+codh`. Цикл очікування відпускання кнопок розташовано в **рядках** 64, 65. У **рядку** 64 викликається підпрограма `incod`. Вона визначає код стану клавіатури й порівнює отриманий код із числом, записаним у буфері `codl+codh`.

Якщо код стану клавіатури дорівнює коду в буфері, то після виходу з підпрограми прапор `Z` буде встановлений. А якщо ні, то— скинутий. Рівність кодів означає, що кнопки відпущені. Тому оператор умовного переходу в **рядку** 65 перевіряє значення прапора `Z`. Поки коди різні, керування передається по мітці `te`, і цикл очікування триває. Як тільки коди виявляться рівними, цикл переривається, і керування переходить до **рядка** 66.

У **рядку** 66 починається **цикл очікування натискання кнопки**. Цикл займає **рядки** 66, 67 і виглядає майже так само, як цикл очікування відпускання. Відмінність полягає в операторові умовного переходу. Замість `brne` (перехід за умовою «не рівно») застосовується оператор `breq` (перехід за умовою «рівно»).

У буфері `codl+codh` як і раніше перебуває код стану клавіатури при повністю відпущених кнопках. Тому вихід з даного циклу відбудеться тоді, коли буде натиснута кожна із кнопок (SI-S10).

Як тільки натискання буде виявлено, програма переходить у наступну стадію. Отриманий код стану клавіатури повинен стати першим кодом ключової комбінації. Але перш ніж починати цикл уведення цієї комбінації, програма виконує дві дуже важливі операції: У **рядках** 68, 69 у реєстрову пару `Z` записується адреса початку буфера в ОЗУ, куди буде міститися комбінація, що вводиться. Регістр `Z` буде зберігати поточний показчик цього буфера.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 77</i>

Друга важлива операція проводиться в **рядку 70**. Отут стає 0 лічильник байтів, записаних у буфер. Після цього починається цикл введення кодової комбінації. Цикл займає **рядки 71–88**. Починається робота циклу з формування захисної затримки. Чому починається із затримки, якщо ми тільки що одержали перше натискання кнопки?

А згідно з алгоритмом після натискання покладено формувати затримку. Для формування захисної затримки використовується підпрограма wait, що працює в режимі 1. Спочатку в **рядку 72** у регістр data записується номер режиму. Потім у **рядку 73** викликається підпрограма wait.

Після закінчення захисної затримки в **рядку 74** знову проводиться введення коду стану. У **рядках 75, 76** отриманий код записується в буфер. Для запису використовуються команди, що збільшують значення покажчика (Z). Тому після запису кожного чергового байта покажчик пересувається в наступну позицію. Потім у **рядках 77, 78** збільшується значення лічильника прийнятих байтів. Тому що ми записали два байти, те й значення лічильника збільшується двічі.

У **рядках 79, 80** проводиться оцінка довжини введеного коду. Якщо довжина перевищить розміри буфера, то цикл введення коду достроково припиняється. У **рядку 79** проводиться порівняння поточного значення лічильника з розміром буфера.

У **рядку 80** перебуває оператор умовного переходу, який передає керування по мітці t7 у випадку, якщо довжина коду перевищить розмір буфера. У **рядках 81, 82** код стану клавіатури записується в буфер codh-codl. Робиться це для того, щоб наступний уведений код стану можна було порівнювати з поточним.

Далі програма повинна **очікувати чергова зміни коду стану**. Але спочатку потрібно запустити таймер, щоб він почав формування захисного проміжку часу. Якщо протягом цього проміжку не буде натиснута жодна кнопка, то це повинне послужити сигналом до виходу із циклу введення ключової комбінації. Запуск таймера

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 78</i>

проводиться за допомогою підпрограми **wait** у режимі номер два. У **рядку 83** у реєстр **data** записується номер режиму, а в **рядку 84** викликається сама підпрограма.

У **рядках 85–88** організований комбінований цикл очікування. У тілі циклу відбувається відразу кілька операцій. По-перше, вводиться нове значення коду стану клавіатури (**рядок 85**). У процесі введення новий код рівняється зі старим, який зберігається в буфері `codh-codl`. Якщо коди не рівні, то оператор умовного переходу в **рядку 86** передає керування по мітці `m3`, де відбувається формування захисної затримки, потім повторне зчитування й запис коду в буфер і так далі.

Якщо нове значення коду рівно старому, комбінований цикл триває. У **рядку 87** проводиться перевірка прапора затримки `flz`. Якщо прапор дорівнює нулю, це значить, що захисний проміжок часу ще не закінчився. У цьому випадку оператор умовного переходу в **рядку 88** передає керування по мітці `m6`, і комбінований цикл триває спочатку. Якщо значення прапора `flz` дорівнює одиниці, то цикл завершується, і керування переходить до **рядка 89**.

У **рядках 89, 90** відбувається перевірка перемикача режимів роботи (SII). Залежно від стану цього перемикача отримана тільки що кодова комбінація або записується в EEPROM (режим «Запис»), або надходить у процедуру перевірки (режим «Робота»). Команда збіс у **рядку 89** перевіряє значення сьомого біта реєстру PINB.

Якщо біт дорівнює нулю (контакти тумблера замкнені), то **рядок 90** не виконується, і керування переходить до **рядка 91**, де починається процедура запису в EEPROM. Якщо значення розряду дорівнює одиниці (контакти тумблера не замкнені), оператор умовного переходу в **рядку 90** передає керування по мітці `m9` на початок процедури порівняння.

Процедура запису ключової комбінації в EEPROM

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 79</i>

Ця процедура займає **рядки 91–101**. До початку цієї процедури кодова комбінація вже перебуває в буфері ОЗУ. Довжина комбінації втримується в змінній count. Нам залишається тільки записати все це в EEPROM.

У **рядках 91–93** в EEPROM записується довжина комбінації. У якості адреси для запису використовується мітка klen. Ця мітка вказує на гніздо, яке спеціально зарезервована для цієї мети (див. **рядок 20**). Для запису байта в EEPROM використовується підпрограма eewr.

У **рядку 91** довжина комбінації міститься в реєстр data. У **рядку 92** адреса міститься в реєстр addr. Потім викликається підпрограма eewr (**рядок 93**).

У **рядках 97–100** розташований цикл запису всіх байтів ключової комбінації. Перед початком циклу в реєстр addr записується адреса першого гнізда буфера-приймача, що перебуває в EEPROM (**рядок 94**). А в реєстрову пару Z записується адреса першого гнізда буфера-джерела, що перебуває в ОЗУ (**рядка 95, 96**).

У процесі запису ключової комбінації реєстр count використовується для підрахунку записаних байтів. На початку, як уже говорилося, він містить довжину комбінації. При записі кожного байта вміст count зменшується. Коли воно виявиться рівним нулю, цикл запису припиняється.

Цикл починається з того, що черговий байт ключової комбінації, що перебуває в ОЗУ, міститься в реєстр data (**рядок 97**). Нагадаю, що адреса вже перебуває в реєстрі addr. У **рядку 98** викликається підпрограма, яка записує байт в EEPROM. Та ж підпрограма збільшує значення addr на одиницю. У **рядку 99** зменшується значення реєстру count.

Перевірку вмісту count на рівність нулю робить оператор brne у **рядку 100**. Якщо вміст не дорівнює нулю, то оператор передає керування на мітку m8, і цикл запису ключової комбінації триває. А

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 80</i>

якщо ні, то цикл завершується, і керування переходить до **рядка 101**. Тобто до процедури відкривання замка.

Тепер розберемося, навіщо після запису коду викликається процедура відкривання замка. Це зроблене для зручності. Після введення кодової комбінації необхідно витримати паузу в 1 із для того, щоб уведена комбінація записалася в EEPROM. Для того, щоб точно знати, коли закінчується ця пауза, використовується спрацьовування замка. Як тільки клацне соленоїд, можна вважати, що введення закінчене.

Процедура перевірки кода

Ця процедура займає **рядки 102–114**. Процедура відкривання замка багато в чому схожа на процедуру запису. Для читання байта з EEPROM використовується підпрограма eerd. Перед викликом цієї підпрограми адреса гнізда, звідки буде прочитана інформація, записується в регістр `addr`. Прочитане з EEPROM значення вертається в регістрі `data`.

У **рядках 102–105** відбувається зчитування довжини послідовності з EEPROM і порівняння її з довжиною нової послідовності. Спочатку в **рядку 102** адреса гнізда, де зберігається довжина коду, записується в `addr`. Потім викликається підпрограма `eerd` (**рядок 103**).

У **рядку 104** рівняється отримане з EEPROM значення (регістр `data`) і нове значення довжини коду (регістр `count`). У випадку нерівності цих двох величин оператор умовного переходу в **рядку 105** передає керування в початок програми. У цьому випадку подальша перевірка більше не проводиться. Двері залишається закритої.

Якщо довжина коду виявилася правильною, починається цикл побайтново перевірки кодів. Спочатку в регістр `addr` записується початкова адреса буфера в EEPROM (**рядок 106**). У регістрову пару **Z** записується початкова адреса буфера в ОЗУ (**рядка 107, 108**). Сам цикл порівняння займає **рядки 109–114**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 81</i>

У **рядку 109** викликається підпрограма, яка читає черговий байт із EEPROM. Байт міститься в реєстр data. У **рядку 110** читається байт із ОЗУ. Цей байт міститься в реєстр temp. У **рядку 111** ці два байти рівняються. Якщо байти не рівні, оператор умовного переходу в **рядку 112** перериває процес порівняння й передає керування на початок програми. Тобто двері й у цьому випадку залишається закритої.

Якщо байти рівні, виконується зменшення вмісту лічильника count (**рядок 113**). Якщо після зменшення вміст count ще не досяглося нуля, керування передається по мітці mlo (оператор brne у **рядку 114**), і цикл порівняння триває. Коли вміст count виявиться рівним нулю, процес порівняння закінчується. Це означає, що всі байти обох версій ключової комбінації виявилися рівні. Тому програма плавно переходить до процедури відкривання замка (до рядка 115).

Процедура відкривання замка

Ця процедура займає **рядки 115–119**. Процедура дуже проста. Для перевірки коду на четвертий розряд порту PB подається одиничний сигнал, який відкриває транзистор ключа VT1 (див. **рис. 5.1**). Реле спрацьовує, і замок відкривається. Подавши відкриваючий сигнал, програма витримує паузу, а потім сигнал знімає. Після цього замок закривається. Тривалість паузи рівна однієї секунді. Цього часу досить, щоб відкрити двері.

Подача відкриваючого сигналу на вихід здійснюється в **рядку 115**. У **рядках 116, 117** проводиться виклик процедури затримки. При цьому вибирається режим номер 3. Спочатку в реєстр data міститься код режиму затримки (**рядок 116**). Потім викликається підпрограма wait (**рядок 117**). У **рядку 118** знімається сигнал відкривання двері. У **рядку 119** процедура відкривання замка завершується. Оператор безумовного переходу, що перебуває в цьому рядку, передає керування на початок програми. І весь процес починається спочатку.

2.4. Порядок виконання роботи

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 82</i>

1. Створити новий проект в середовищі PROTEUS. Скласти схему кодового замка згідно рис. 5.1.

2. Створити новий проект в AVR Studio, вибрати New Project, далі вибрати як налагоджувальну платформу **Atmel AVR Assembler, AVR Simulator** і мікросхему **ATtiny2313**.

3. Для мікроконтролера ATtiny2313 написати програму кодового замка на 10 кнопок для введення коду, позначених цифрами від «0» до «9». Замок повинен мати перемикач режимів «Запис/Робота». У випадку правильного набору коду замок повинен включати виконавчий механізм замка (соленоїд або електромагнітну засувку). Введення коду повинно проводитися описаним в теретичних відомостях способом.

Для створення програми користуватися інформацією з теоретичних відомостей та лістингом шаблону 5.1.

4. Виконати відладку програму в AVR Studio в покроковому режимі за допомогою кнопки **F11**.

5. Відкрити створений в PROTEUS проект і підключити написану програму до мікропроцесора. Запустити симуляцію, перевірити працездатність схеми.

6. На основі першої програми створити програму кодового замка на 8 цифр від «0» до «7» і у якого захисна затримка дорівнює 100 мс.

7. Відкрити створений в PROTEUS проект і підключити написану програму до мікропроцесора. Запустити симуляцію, перевірити працездатність схеми.

8. Написати звіт. Зробити висновки.

1.5. Зміст звіту. Звіт повинен містити:

1. Тексти написаних програм.

2. Перелік використаних команд з поясненням їх призначення.

3. Зробити висновки: про об'єм виконаної роботи, чи досягнена мета роботи, чи практичні результати співпали з теоретичними.

1.6. Контрольні тестові питання.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 83</i>

1. Які мікроконтролери підтримує емулятор електронних пристроїв PROTEUS?

2. Що відображає панель **DEVICES**?

3. Як виконати подачу живлення або певного сигналу в довільну точку схеми?

4. Який вигляд мають файли прошивки виконані на мовах с, assembler, та в машинних кодах?

5. Які дії необхідно провести, щоб проемувати в PROTEUS роботу мікроконтролера?

6. Як задати частоту тактування МК при емуляції?

7. Яке призначення має віртуальний термінал?

8. Як задати необхідний час емуляції?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 84</i>

Практична робота №5

Дослідження функціональних можливостей таймерів

5.1. Навчальні питання

1. Дослідження принципів побудови таймерів.
2. Дослідження способів програмування таймерів та їх використання для керування периферійними пристроями.

5.2. Навчальна мета

Практичне дослідження функціональних можливостей таймерів та методики створення програм з їх використання за допомогою програми AVR Studio та Proteus.

5.3. Теоретичні відомості

Використання таймера. Постановка завдання. В попередніх прикладах для формування затримки ми використали один або декілька вкладених програмних циклів. Однак такий спосіб прийнятний далеко не завжди. Головний недолік подібного методу полягає в тому, що він повністю завантажує центральний процесор. Поки мікроконтроллер зайнятий формуванням затримки, він не може виконувати жодних інших завдань.

Ще один недолік – неможливо з достатньою точністю вибрати час затримки. Набагато кращі результати дає інший спосіб – формування інтервалів часу при допомогою одного із вбудованих таймерів / лічильників мікроконтролера. Будь-який з таймерів / лічильників може працювати як з використанням переривань, так і без переривань. Далі ми розглянемо обидва ці варіанти. І почнемо ми з простішого випадку.

Отже, заново сформулюємо нашу задачу:

Доопрацювати програму «Біжучі вогні», змінивши процедуру формування затримки. Нова процедура повинна використовувати один з внутрішніх таймерів / лічильників і не використовувати переривання.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 85</i>

Схема

Так як ми розробляємо не самостійний пристрій, а лише вдосконалене керуючу програму, то схема пристрою не змінюється.

Алгоритм

Як відомо, в мікроконтролері ATtiny2313 є два вбудованих таймера-лічильника. Тому спочатку нам потрібно вибрати, який з них ми будемо використовувати. Виходити будемо з заданого часу затримки 200 мс. Як відомо, для формування тимчасових інтервалів таймер / лічильник просто підраховує тактові імпульси від системного генератора.

Частота сигналу цього генератора в нашому випадку дорівнює 4 МГц. А період імпульсів $1/4 = 0,25$ мкс. Для того, щоб отримати на виході 200 мс, необхідно мати коефіцієнт рахунку або ділення тактової частоти з рахунком до одного, рівний $200 \cdot 10^{-3} / 0,25 \cdot 10^{-6} = 800 \cdot 10^3$ (вісімсот тисяч разів).

Мікросхема ATtiny2313 містить два таймери. Один восьмирозрядний і один шістнадцяти. Вісьмирозрядний таймер має максимальний коефіцієнт перерахунку $2^8=256$, а шістнадцятирозрядний – $2^{16} = 65536$. Тобто навіть шістнадцятирозрядного таймера нам не вистачить для формування необхідної затримки. Доведеться скористатися попередніми дільником. Цей дільник проводить попереднє поділ частоти тактового сигналу перед тим, як той надійде на вхід таймера / лічильника.

Програмним шляхом можна вибрати один з чотирьох фіксованих коефіцієнту поділу. Виберемо найбільший можливий коефіцієнт поділу переддільника (1024), рядки 16–17 прикладу 4.1 [4-9]. Тоді на його виході ми отримаємо сигнал з частотою $4\text{МГц}/1024 = 3906$ Гц. Період такого сигналу буде дорівнює $1 / 3906 = 0,256$ мс. Саме цей сигнал надходить на вхід нашого таймера, який забезпечує остаточний поділ. Порахуємо коефіцієнт поділу, який наш таймер

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 86</i>

повинен нам забезпечити: $200 / 0,256$ приблизно = 780. Такий коефіцієнт підрахунку нам може забезпечити тільки таймер T1.

Отже, ми визначилися як з вибором таймера, так і з його налаштуваннями. Тепер можна приступати до створення нової підпрограми затримки. Перш, ніж це зробити, спробуємо описати алгоритм її роботи. Даний алгоритм передбачає, що всі необхідні настройки таймера попереднього дільника проведено до першого виклику підпрограми, таймер запущений і знаходиться в режимі безперервної рахунки.

Алгоритм підпрограми затримки представлений нижче.

1. Записати в рахунковий реєстр таймера T1 нульове значення.
2. Почати цикл перевірки вмісту рахункового реєстра. В тілі циклу програма повинна багаторазово зчитувати вміст рахункового реєстра таймера і перевіряти, чи не досягло воно свого кінцевого значення (тобто значення 780).
3. При досягненні рахунковим реєстром кінцевого значення, завершити цикл перевірки.
4. Вийти з підпрограми затримки.

Програма на Асемблері

Програма «Біжучі вогні» з новим варіантом підпрограми затримки приведена в прикладі 4.1. Нова підпрограма затримки використовує таймер T1 і описаний вище алгоритм роботи. Розглянемо докладніше, як працює така програма. А почнемо, як завжди, з опису нових для нас операторів.

.equ

Псевдооператор присвоєння. Назва оператора походить від англійського слова «еквівалентно» (equality). Використовується для присвоєння імен різним констант. В рядку 5 прикладу 4.1 числу 780 надається ім'я kdel. Тепер у будь-якому місці програми замість числа 780 можна застосовувати константу kdel. Ім'я для константи має те ж значення, що й ім'я для змінної. По-перше, по осмисленому імені легко зрозуміти призначення константи. Наприклад, kdel

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 87</i>

розшифровується, як «коефіцієнт поділу». А, по-друге, це зручно при зміні значення. Поміняйте в рядку 5 число 780, приміром, на 800, і скрізь, де б не зустрілася константа k_{del} , вона вже буде мати нове значення.

срі

Порівняння вмісту РЗП з константою. Ця команда має два параметри. Перший параметр – ім'я регістра загального призначення, вміст якого підлягає порівнянню. Другий параметр – якась константа, з якою порівняється вміст РЗП. За результатами порівняння встановлюються всі прапори регістру SREG. Прапори встановлюються точно також, як якщо б вміст РЗП віднімався з константи. А саме: прапор переносу C встановлюється в тому випадку, якщо при вирахуванні даних чисел виникає перенесення в старший розряд (вміст регістру менше константи), і скидається, якщо немає переносу. Прапор Z (нульовий результат) встановлюється при рівності вмісту РЗП і константи і скидається в разі їх нерівності. Все інші прапори встановлюються відповідно до свого призначення. Після того, як значення прапорів визначені, вони можуть бути використані різними умовними операторами. В рядку 44 програми (приклад 4.1) оператор **срі** виробляє порівняння вмісту регістра `temp` з числом `0xD0`.

brlo

Перехід за умовою «менше». Мається на увазі, що в попередній команді, в результаті порівняння (чи вирахування) двох операндів, перший операнд виявився менше, ніж другий. Для перевірки цієї умови оператор використовує прапор переносу C. Перехід відбувається лише в тому випадку, якщо $C = 1$. В рядку 45 програми умовний перехід використовується для того, щоб передати управління на рядок з міткою **wt1** в тому випадку, якщо за результатами попереднього порівняння (рядок 44) виявилось, що вміст регістра `temp` менше, ніж `0xD0`.

Опис програми (приклад 5.1)

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 88</i>

Основна відмінність нової програми – повна переробка підпрограми затримки. В зв'язку з тим, що нова підпрограма затримки використовує таймер, для її нормальної роботи довелося також доопрацювати модуль ініціалізації основної програми.

Перша доробка модуля ініціалізації – команда в рядку 5. Ця команда описує константу `kdel`, тобто коефіцієнта ділення тактової частоти МП для таймера. Зверніть увагу, що значення цієї константи 780. Якщо перевести це значення в двійкову форму, то кількість розрядів такого числа буде більше восьми. А це значить, що для представлення константи в двійковому вигляді буде потрібно не менше двох байтів.

Далі в програмі з цією константою порівнюється вміст рахункового реєстра таймера `TI`, який теж має шістнадцять розрядів. Однак мікроконтролери AVR працюють лише з восьмирозрядними величинами. Рахунковий реєстр таймера `TI` являє собою два восьмирозрядних реєстру `TCNT1L` і `TCNT1H`. Використовуваний для порівняння оператор `brlo` також працює з восьмирозрядними величинами. Як же виконується таке порівняння?

Порівняння відбувається в два етапи. Спочатку порівнюються молодші розряди обох величин, потім старші. Молодше і старше значення рахункового реєстра зберігаються у двох відповідних реєстрах `TCNT1L` і `TCNT1H`. А для виділення молодшого та старшого байта константи в мові Асемблер існують спеціальні функції `low` і `high`. Дані функції використовуються в рядках 44 і 47 програми.

Наступна доробка модуля ініціалізації – це дві команди, які вибирають режим роботи таймера (рядки 16,17). Ці команди записують в реєстр `TCCR1B` константу `0x05` (`0b00000101`). В якості допоміжного реєстра використовується `temp`. Реєстр `TCCR1B` – це один із двох реєстрів вибору режимів роботи таймера `TI`. При запису коду `0x05` в цей реєстр встановлюється коефіцієнт попереднього ділення $1 / 1024$, і таймер переходить в режим рахунку. Другий реєстр конфігурації таймера називається `TCCR1A`. Його значення потрібно оставити за замовчуванням. Детальніше про реєстрах і режимах роботи

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 89</i>

таймера дивитися в главі б. Останні зміни основної частини програми торкнулися команд виклику підпрограми затримки. Виклик затримки відбувається в рядках 27 і 35. Інших змін основної частини програми не було потрібно. Нова підпрограма затримки займає рядки 39–50. Починається підпрограма традиційно збереженням вмісту всіх використовуваних нею регістрів. В даному випадку треба зберегти лише вміст одного регістра temp (див. рядок 39). Наступні три команди виробляють запис нульового значення в рахунковий регістр таймера TI. Спочатку нуль записується в регістр temp (рядок 40). А потім вміст temp по черзі поміщається в регістри TCNT1H і TCNT1L (рядки 41,42).

Порядок запису інформації в пару регістрів TCNT1H, TCNT1L не випадковий. Ці два регістри об'єднують властивості так званої подвійної буферизації. Правила роботи з такими регістрами вимагають, щоб при записі значення в ці регістри спочатку записувався старший регістр TCNT1H, а потім молодший TCNT1L. Справа в тому, що при записі старшого байта в регістр TCNT1H він не потрапляє відразу за призначенням, а зберігається в спеціальному внутрішньому регістрі. Коли ж поступає команда запису молодшого байта в регістр TCNT1L, обидва байти записуються одночасно. В цьому і полягає подвійна буферизація. Використання двійкової буферизації дозволяє змінювати значення рахункового регістра на ходу, не зупиняючи таймера.

Приклад 5.1

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 90

```

#####
;##          Пример 6          ##
;##          "Бегущие огни"    ##
;##          с использованием таймера ##
;#####
;----- Псевдокоманды управления
1  .include "tn2313def.inc"      ; Присоединение файла описаний
2  .list                        ; Включение листинга
3  .def temp = R16              ; Определение главного рабочего регистра
4  .def rat = R20               ; Определение рабочего регистра для команд сдвига
5  .equ kdel = 780

;----- Начало программного кода
6  .cseg                       ; Выбор сегмента программного кода
7  .org 0                      ; Установка текущего адреса на ноль

;----- Инициализация стека
8  ldi temp, RAMEND            ; Выбор адреса вершины стека
9  out SPL, temp               ; Запись его в регистр стека

;----- Инициализация портов ВВ
12 ldi temp, 0                 ; Записываем ноль в регистр temp
11 out DDRD, temp              ; Записываем этот ноль в DDRD (порт PD на ввод)
12 ldi temp, 0xFF              ; Записываем число $FF в регистр temp
13 out DDRB, temp              ; Записываем temp в DDRB (порт PB на вывод)
14 out PORTB, temp             ; Записываем temp в PORTB (потушить светодиод)
15 out PORTD, temp             ; Записываем temp в PORTD (включаем внутр.резист.)

;----- Инициализация таймера T1
16 ldi temp, 0x05              ; Код конфигурации записываем в temp
17 out TCCR1B, temp            ; Переносим его в регистр конфигурации таймера

;----- Инициализация компаратора
18 ldi temp, 0x80              ; Выключение компаратора
19 out ACSR, temp

;----- Начало основного цикла
20 main: in temp, PIND          ; Читаем содержимое порта PD
21 sbrc temp, 0                ; Проверка младшего разряда

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 91

```

22      rjmp      n3          ; Если не ноль, переходим в начало
;----- Сдвиг вправо
23      n1:      ldi          rab, 0b10000000 ; Запись начального значения
24      n2:      ldi          temp, 0xFF
25      eor      temp, rab      ; Инверсия содержимого регистра rab
26      out      PORTB, rab     ; Вывод текущего значения в порт PB
27      rcall   wait1         ; Задержка
28      lsr      rab           ; Сдвиг содержимого рабочего регистра
29      brcc   m2             ; Если не дошло до конца регистра продолжить
30      rjmp      main        ; На начало
;----- Сдвиг влево
31      n3:      ldi          rab, 0b00000001 ; Запись начального значения
32      n4:      ldi          temp, 0xFF
33      eor      temp, rab      ; Инверсия содержимого регистра rab
34      out      PORTB, rab     ; Вывод текущего значения в порт PB
35      rcall   wait1         ; Задержка
36      lsl      rab           ; Сдвиг содержимого рабочего регистра
37      brcc   m4             ; Если не дошло до конца регистра продолжить
38      rjmp      main        ; На начало
;----- Подпрограмма задержки
39      wait1:   push        temp      ; Сохраняем содержимое регистра temp
40      ldi      temp, 0           ; Помещаем temp ноль
41      out      TCNT1H, temp      ; Записываем этот ноль в старший регистр таймера
42      out      TCNT1L, temp      ; Записываем этот ноль в младший регистр таймера
43      w1:      in          temp, TCNT1L ; Чтение младшей части счетного регистра
44      cpi      temp, low(kdel) ; Сравнение с числом $0C
45      brlo   w1              ; Переход, если temp меньше чем kdel
46      in      temp, TCNT1H ; Чтение старшей части счетного регистра
47      cpi      temp, high(kdel) ; Сравнение с числом $03
48      brlo   w1              ; Переход, если temp меньше чем $03
49      pop     temp           ; Восстанавливаем значение регистра temp
50      ret

```

Після запису нульового значення в рахунковий реєстр починається цикл перевірки. Він займає рядки 43–48 програми. Порівняння відбувається в два етапи. В рядках 43–45 порівнюється молодша частина рахункового реєстра <молодшим байтом коефіцієнта ділення. В рядках 46–48 порівнюються старші байти. Розглянемо це докладніше.

В рядку 43 вміст реєстра TCNT1L міститься в реєстр temp. В 44 відбувається порівняння вмісту реєстра temp з молодшим байтом константи. Команда умовного переходу в рядку 45 передає керування на початок циклу порівняння лише в тому випадку якщо вміст реєстру ще не досягло необхідного значення.

В рядках 46–48 такі ж операції порівняння проводиться для реєстра TCNT1H. При цьому використовується старший Байга констату kdel. Якщо старші розряд рахункового реєстра не досяг необхідного значення, то управління передається на мітку w1. То є в

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 92

цьому випадку програма знову повторює порівняння молодших розрядів.

Такий порядок також диктується наявністю подвійної буферизації. При читанні молодшої частини регістру старша його частина запам'ятовується в спеціальному внутрішньому буфері. Команда читання старшої частини регістру насправді читає вміст цього буфера.

Поки відбувається цикл порівняння, лічильник знаходиться в режимі рахунку. Вміст рахункового регістра поступово збільшується і, врешті–решт, досягає необхідного значення. Поки відбувається черговий цикл перевірки, вміст рахункового регістра може навіть перевищити значення константи.

В цьому випадку переходів у рядку 45 і в рядку 48 не відбудеться. В результаті підпрограма перейде до свого завершення. В рядку 49 відбувається відновля вмісту регістра temp. А в рядку 50 – вихід з підпрограми.

Принципова схема біжучих вогнів зі світлодіодами наведена на рис. 5.1.

5.4. Порядок виконання роботи

1. Створити новий проект в AVR Studio, при цьому вибрати як налагоджувальну платформу *AVR Simulator* і мікросхему *Atiny2313*.

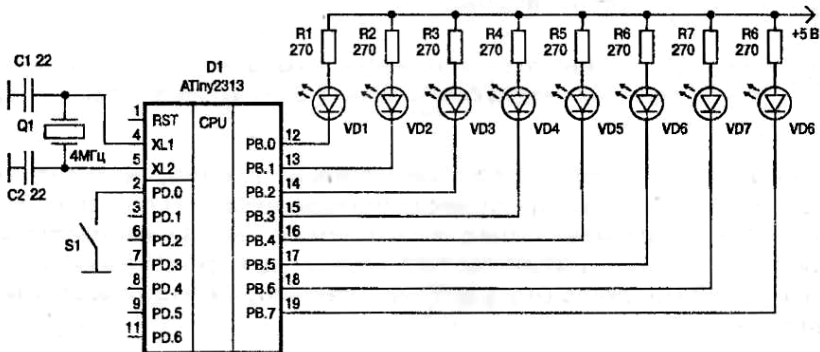


Рис. 5.1. Принципова схема автомата біжучих вогнів

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 93</i>

2. Для мікроконтролера **Atiny2313** на схемі рис. 5.1 написати програму послідовного включення лампочок (світлодіодів) із заданою тривалістю та черговістю світіння.

3. Виконати відладку програму.

4. Створити проект із заданою схемою в середовищі PROTEUS і підключити написану програму для **Atiny2313**. Запустити симуляцію, перевірити працездатність схеми.

5.5. Зміст звіту

1. Тексти написаних програм.

2. Зібрані в Протеусі схеми.

3. До кожної програми структурну схему (flow chart) алгоритмів.

4. Перелік використаних команд з поясненням їх призначення.

5. Зробити висновки: про об'єм виконаної роботи, чи досягнена мета роботи, чи практичні результати співпали з теоретичними.

5.6. Контрольні тестові питання

1. Яка є архітектура МК та таймерів?

2. Яка кількість та параметри таймерів T0–T2?

3. Поясніть принцип роботи таймера T0.

4. Які є режими роботи таймерів?

5. Які принципи програмування таймерів МК AVR?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 94

Практична робота №6

Дослідження функціональних можливостей пам'яті МК

6.1. Навчальні питання

1. Дослідження режимів роботи та функціональних можливостей пам'яті МК.
2. Написання програми для відтворення звуку та моделювання її роботи в середовищі PROTEUS.

6.2. Навчальна мета

Практичне дослідження режимів роботи та функціональних можливостей пам'яті МК.

6.3. Теоретичні відомості

Музична шкатулка. Постановка завдання. І так, починаємо ряд прикладів, у яких буде не тільки представлений опис якихось схем на мікроконтролері, але і описаний процес їх розробки. Всі три приклади, наведені в цій книзі спеціально розроблені автором, і призначені для навчання початківців програмістів і є заключним етапом уроку з програмування.

У першому прикладі ми покажемо процес розробки простого мікропроцесорного пристрою, призначеного для автоматичного відтворення декількох простих мелодій. Причому всі ці мелодії ми жорстко "зашьєм" з вшанування пам'яті мікроконтролера. Зручніше за все для цього використовувати програмну пам'ять контролера. Тобто, ту ж саму пам'ять, куди записується керуюча програма. Можна було б, звичайно, для зберігання мелодій використовувати і EEPROM. Але обсяг EEPROM в мікросхемі ATtiny2313 складає всього 128 байт, а обсяг пам'яті програм – 2 кілобайт. Тому пам'ять програм у даному випадку переважно.

Сформулюємо завдання наступним чином: *«Розробити пристрій, призначений для відтворення простих одноголосих мелодій, записаних в пам'ять програм на етапі програмування. Пристрій повинен мати сім кнопок керування. Кожній з кнопок повинна*

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 95

відповідати своя мелодія. Мелодія відтворюється при натисканні та утриманні кнопки. При відпуску всіх кнопок відтворення мелодій припиняється ».

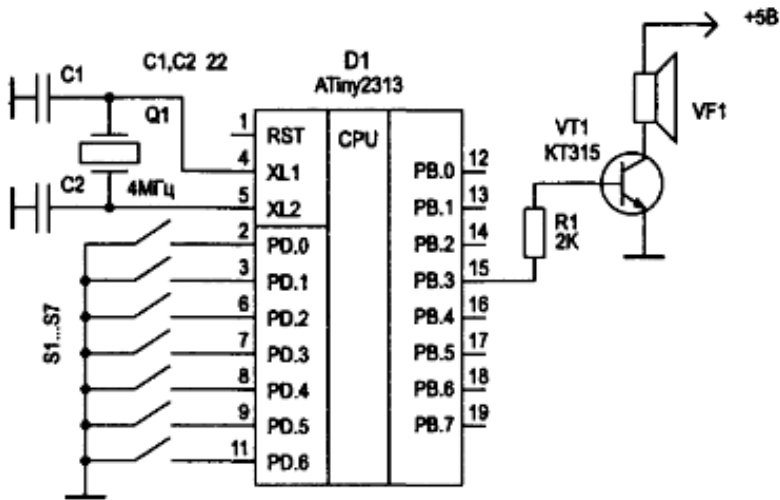


Рис. 6.1. Принципова електрична схема музичної шкатулки

Опис схеми

Принципова електрична схема пристрою, узадовольняюча сформульованим вище вимогам наведена на рис. 6.1. Кнопки S1 ... S7 призначені для вибору мелодій. Для відтворення мелодії використовується звуковий випромінювач VF1, сигнал на який поступає з виходу PB3 мікроконтролера. У як підсилювач сигналу використовується електронний ключ R1, VT1.

Алгоритм

Для початку нам потрібно придумати, як ми будемо зберігати мелодії в пам'яті. Для того, щоб в пам'яті можна було щось зберігати, треба спочатку це щось будь-який спосіб закодувати. Будь-яка мелодія складається з нот. Кожна нота має свій тон (частоту) і тривалість звучання. Для того, щоб закодувати тон ноти, можна

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 96</i>

просто всі ноти пронумерувати по порядку. Зручніше нумерувати, починаючи з найнижчого тону. На клавіатурі клавішного інструменту це буде зліва направо.

Відомо, що весь музичний ряд ділиться на **октави**. Якщо ви думаєте, що в кожній октаві сім нот, то ви погано знаєте фізичні основи музичного ряду. Насправді в сучасному музичному ряду кожна октава ділиться 12 на нот. Сім основних нот (білі клавіші) та п'ять додаткових (чорні клавіші).

Поділ на основні і додаткові ноти склалося історично. В даний час використовується музичний лад, в якому всі 12 нот однієї октави рівнозначні. Частоти будь-яких двох сусідніх нот відрізняються один від одного в однакову кількість разів. При цьому частоти однойменних нот в двох сусідніх октавах відрізняються рівно в два рази. Більш докладно про це ви можете прочитати в [4–9].

Для нас же важливо те, що коди всіх цих нотах ми повинні привласнювати в порядку зростання частоти, і почнемо ми з ноти «До» першої октави. Для музичної скриньки більш низькі ноти не потрібні.

В табл. 6.1 показані коди для всієї першої октави. Наступна, друга октава продовжує перший і за кодуванням, і по набору частот. Так нота «До» другої октави буде мати код 13, а частоту $f_{12}=f_0 \times 2$. А нота «Ре» другої октави буде мати код 14 і частоту $f_{12}=f_1 \times 2$. І так далі.

Таблиця 6.1

Кодування нот першої октави

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 97</i>

Код	Нота	Частота	Код	Нота	Частота
1	До	f_0	7	Фа#	$f_6=f_5/K$
2	До#	$f_1=f_0/K$	8	Соль	$f_7=f_6/K$
3	Ре	$f_2=f_1/K$	9	Соль#	$f_8=f_7/K$
4	Ре#	$f_3=f_2/K$	10	Ля	$f_9=f_8/K$
5	Ми	$f_4=f_3/K$	11	Ля#	$f_{10}=f_9/K$
6	Фа	$f_5=f_4/K$	12	Си	$f_{11}=f_{10}/K$

Для справки: $K = \sqrt[12]{2}$

Музична тривалість теж легко кодується. У музиці застосовують не довільну тривалість, а тривалість, виражену частками від цілої (див. табл. 7.2). В залежності від темпу реальна тривалість цілої ноти змінюється. Для збереження мелодії необхідно дотримуватися лише співвідношення між тривалістю. Тому нам необхідно закодувати лише сім варіантів тривалості. Привласнимо їм коди від 0 до 6. Наприклад так, як це показано у графі «Код» табл. 2. Призначення графі «Коефіцієнт ділення» ми поки що опустимо.

Таблиця 6.2

Кодування музичних тривалостей

Код	Длительность	Коефициент деления
0	1 (целая)	64
1	1/2 (половинная)	128
2	1/4 (четверть)	256
3	1/8 (восьмая)	512
4	1/16 (шестнадцатая)	1024
5	1/32 (тридцать вторая)	2048
6	1/64 (шестьдесят четвертая)	4096

Крім нот, будь-яка мелодія обов'язково містить музичні паузи.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 98

Визначення. Паузи – це проміжки часу, коли ні один звук не звучить. Загальна тривалість музичних пауз приймає точно такі ж значення, як і тривалість нот.

У зв'язку з цим зручно представити паузу як ще одну ноту. Ноту без звуку. Такий ноті логічно привласнити нульовий код.

Кодуємо мелодії

Для економії пам'яті зручніше кожен ноту кодувати одним байтом. Домовимося, що три старших біта ми будемо використовувати для кодування тривалості ноти, а решта п'ять бітів – для кодування її тону. П'ятьма бітами можна закодувати до 32 різних нот, що цілком вистачить для музичної скриньки.

Отже, якщо використовувати наведений вище спосіб кодування, то код ноти ля першої октави тривалістю 1/4 в двійковому вигляді буде рівний, рис. 7.2.

$$\underline{01001001} = 73$$

Рис. 6.2. Приклад кодування нот

Тепер ми можемо приступати до **кодування мелодій**. Для того, щоб закодувати мелодію, нам потрібна її нотний запис. Використовуючи нотний запис, ми повинні привласнити кожній ноті і кожній музичної паузи свій **код**.

Ланцюжок таких кодів і буде являти собою закодовану мелодію. За умовами завдання наша музична скриня повинна вміти відтворювати сім різних мелодій. Коди всіх семи мелодій ми розмістимо в програмній пам'яті мікроконтролера.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 99</i>

Як визначити **кінець кожній мелодії**? Для того, щоб комп'ютер знав, де закінчується кожна мелодія, використовуємо код 255 як ознака кінця.

Тепер нам потрібно придумати, як мікроконтроллер буде знаходити **початок кожної мелодії**. Всі мелодії мають різну довжину, а в пам'яті вони будуть записані одна за одною. Тому адресу початку кожної мелодії залежить від довжини всіх попередніх. Зручніше за все просто за фактом визначити адресу початку кожної мелодії і помістити всі сім адрес в спеціальну таблицю.

Крім цієї таблиці нам ще знадобиться таблиця коефіцієнтів розподілу для всіх 32 нот і таблиця, що зберігає константи затримки для всіх використовуваних нами музичних тривалостей.

Алгоритм роботи музичної скриньки

Тепер ми можемо сформулювати алгоритм роботи музичної скриньки.

1. Просканувати клавіатуру і визначити номер найпершої натиснутою клавіші.
2. Витягти з таблиці почав мелодій значення елемента, номер якого відповідає тільки що певним номером натиснутою клавіші. Це значення дорівнюватиме адресою в програмній пам'яті, де починається потрібна нам мелодія.
3. Почати цикл відтворення мелодії. Для цього по черзі витягати коди нот з пам'яті, починаючи з адреси, на яку ми визначили в пункті 2 алгоритму.
4. Кожен код ноти розкласти на код тону і код тривалості.
5. Якщо код тону дорівнює нулю, вимкнути звук і перейти до формування затримки (до п. 9 цього алгоритму).
6. Якщо код тону не дорівнює нулю, витягти з таблиці коефіцієнтів ділення значення елемента з номером, рівним коду тону.
7. Записати коефіцієнт розподілу, який ми знайшли в пункті 6 цього алгоритму, у реєстр збігу таймера T1.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 100</i>

8. Увімкнути звук (підключити висновок ОС1А до виходу таймера Т1).
9. Витягти з таблиці тривалостей затримки значення елемента з номером, рівним коду тривалості.
10. Сформувати паузу з використанням константи затримки, яку ми знайшли в пункті 9 цього алгоритму.
11. Після закінчення паузи вимкнути звук (відключити ОС1А від виходу таймера),
12. Повторювати цикл (пункти 4–11 цього алгоритму) до тих пір, поки натиснута відповідна кнопка.
13. Якщо черговий код ноти виявиться рівним 255, перейти на початок поточного мелодії, тобто повернутися до п. 3 цього алгоритму.

Програма на Асемблері

Можливий варіант програми на мові Асемблер приведений в прикладі 6.1. Приводиться нижче опис програми розраховано на читачів, знайомих з мовою Асемблера. Якщо ви не знаєте цієї мови, рекомендую звернутися до [4], де представлений докладний урок для початківців.

Приклад 6.1.

```
.include "tn2313def.inc"
.list
.def loop1=R0
.def loop2=R1
.def loop=R21
.def temp=R16
.def temp1=R17
.def count=R18
.def fnota=R19
.def dnota=R20
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 101</i>

```
.cseg
.org 0
START :
rjmp Init
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
reti
```

```
;*****Модуль ініціалізації *****
Init:
```

```
ldi temp,ramend ; ініціалізація стека,
out SPL, temp ;встановлюємо розміри стеку,
;вказуємо адрес останньої комірки ОЗП, яка буде кінцевою
адресою стеку (дно стеку)
```

```
;-----
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 102</i>

```
; Ініціалізація портів введення/виведення
```

```
;-----
```

```
ldi temp,0x08
out PORTB,temp
OUT DDRB ,TEMP
```

```
ldi temp,0x3
out PORTD,temp
ldi temp,0x00
out DDRD,temp
```

```
;-----
```

```
;вимкнення компаратора
```

```
;-----
```

```
ldi temp,0x80
out ACSR,temp
```

```
;-----
```

```
; ініціалізація таймера T1
```

```
;-----
```

```
m1:ldi temp,0x09
out TCCR1B,temp
ldi temp,0x00
out TCCR1A,temp
```

```
.*****
;
```

```
****
```

```
;* Початок основної програми *
```

```
.*****
;
```

```
****
```

```
MAIN:
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 103</i>

```

;-----
;Сканування клавіатури, оцінка номера натисненої кнопки
;-----

```

```

CLR COUNT
IN TEMP,PIND
M2:lsr temp
brcc m3
inc count
cpi count,2
brne m2
rjmp m1

```

```

;-----
;Вибір мелодії
;-----

```

```

m3:mov YL, count
ldi ZL,low(tabm*2)
ldi ZH,high(tabm*2)
rcall addw

lpm XL,Z+
lpm XH,Z

```

```

;-----
;Відтворення мелодії
;-----

```

```

m4:mov ZH,XH
mov ZL,XL

```

```

m5:in temp, PIND
cpi temp, 0x3

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 104</i>

```

breq m1
lpm temp,Z
cpi temp,0xFF
breq m4

```

```

andi temp,0x1F
mov fnota,temp
lpm temp,Z+
rol temp
rol temp
rol temp
rol temp
andi temp,0x07
mov dnota,temp
rcall nota
rjmp m5

```

```

;*****

```

```

****

```

```

;*          Допоміжні підпрограми          *
;*****

```

```

****

```

```

;-----
; Підпрограма 16-розрядного додавання
;-----

```

```

addw: push YH
      lsl  YL
      ldi  YH,0
      add  ZL,YL
      adc  ZH,YH
      pop  YH
      ret

```


Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 105</i>

;Підпрограма виконання однієї ноти

```

nota: push ZH
      push ZL
          push YL
          push temp
          cpi fnota,0x00
          breq nt1
          mov YL,fnota
          ldi ZL,low(tabkd*2)
          ldi ZH,high(tabkd*2)
          rcall addw
          lpm temp,Z+
          lpm temp1,Z
          out OCR1AH,temp1
          out OCR1AL,temp
          ldi temp,0x40
          out TCCR1A,temp

```

```

nt1: rcall wait
      ldi temp,0x00
          out TCCR1A,temp
          ldi dnota,0
          rcall wait
          pop temp
          pop YL
          pop ZL
          pop ZH
          ret

```

;Підпрограма формування затримки

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 106</i>

;-----

```
wait: push ZH
      push ZL
          push YH
          push YL
          mov YL,dnota
          ldi ZL,low(tabz*2)
          ldi ZH,high(tabz*2)
          rcall addw
          lpm YL,Z+
          lpm YH,Z
          clr ZL
          clr ZH
```

;-----

;цикл затримки

;-----

```
w1: ldi loop,255
w2: dec loop
    brne w2
        adiw R30,1
        cp YL,ZL
        brne w1
        cp YH,ZH
        brne w1
    pop YL
    pop YH
    pop ZL
    pop ZH
    ret
```

;-----

;таблиця коефіцієнтів затримки для формування музичної тривалості звучання нот

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 107</i>

;-
tabz: .dw 128,256,512,1024,2048,4096,8192

;-
;таблиця коеф. ділення для всіх нот Di(Finote=Fzad/Di)

;-
tabkd: .dw 0 ;коєфіцієнт ділення для паузи, 0-ї ноти
.dw

9496,8960,8456,7984,7536,7112,6712,6336,5980,5644,5328,5028
.dw

4748,4480,4228,3992,3768,3556,3356,3168,2990,2822,2664,2514
.dw 2374,2240,2114,1996,1884,1778,1678,1584

;-
; таблиця початків усіх мелодій, кожна мітка містить адресу
початку мелодії

;-
tabm: .dw mel1*2, mel2*2

;-
;таблиці мелодій, значення кожної мітки - це адреса початку
певної мелодії

;-

;=====Пісня з кінофільму "Веселі хлопці"=====

mel1: .db 105,109,112,149,116,64,80,148,114,64
.db 78,146,112,96,105,105,109,144,111,64
.db 80,145,112,64,81,178,96,117,117,117
.db 149,116,64,82,146,112,64,79,146,144
.db 96,105,105,107,141,108,109,112,110,102
.db 104,137,128,96,105,105,105,137,102,64
.db 73,142,105,107,109,64,75,137,96,105

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 108</i>

.db 105,105,137,102,105,142,112,64,82,180
.db 96,116,116,116,148,114,112,142,109,64
.db 78,146,144,96,105, 105,107,141,108,109
.db 112,110,102,104,169,96,96,255

;=====Посмішка=====

mel2: .db 107,104,141,139,102,105,104,102,164,128
.db 104,107,109,109,109,111,114,112,111,109
.db 144,139,128,109,111,144,96,111,109,104
.db 107,105,173,128,111,109,112,107,111,109
.db 109,107,102,104,134,132,128,100,103,107
.db 107,107,107,139,112,100,103,102,102,102
.db 134,102,103,107,105,107,108,108,108,108
.db 107,105,107,108,144,142,128,112,107,110
.db 140,112,105,108,107,107,107,105,140,139
.db 139,112,103,102,103,105,108,107,105,103
.db 128,112,107,110,108,108,108,140,112,105
.db 108,107,107,107,139,112,103,102,103,105
.db 108,107,105,103,105,139,132,128,96,96
.db 96,255

Опис програми (приклад 6.1)

Опис програми зручніше почати з кінця. Починаючи з рядка 136 програми розташовується опис так званих таблиць даних. Насправді кожна із цих "таблиць" представляє собою ланцюжок кодів, записуваних у програмну пам'ять мікроконтролера й призначених для кодування того або іншого виду даних. Для опису цих даних використовуються як оператори db, так і оператори dw.

Перша таблиця містить коефіцієнти затримки для формування **всіх** варіантів музичної тривалості. Таблиця починається з адреси,

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 109</i>

відповідного до мітки tabz. Уся таблиця займає один **рядок** програми (**рядок 136**). Тому що в нашій програмі ми будемо застосовувати лише сім варіантів тривалості, таблиця має 7 елементів. Кожний елемент записується на згадку як двухбайтне слово.

У **рядках 137–140** описується **таблиця коефіцієнтів розподілу для Всіх нот**. Початок таблиці відповідає мітці tabkd. Кожний елемент цієї таблиці також має розмір у два байти. Перший елемент таблиці дорівнює нулю. Це невикористований елемент. Ноти номер нуль у нас не існує. Нуль ми використовували для кодування паузи.

У паузі не формується звуковий сигнал, тому й коефіцієнт розподілу там не має значення. Тому значення нульового елемента масиву несуттєво. Опис таблиці розбитий на рядки. Для зручності кожний рядок описує коефіцієнти розподілу для однієї октави. Нульова нота виділена в окремий рядок. Остання октава неповна, тому що наша скринька буде використовувати всього 32 ноти.

У **рядках 143–200** описана **таблиця мелодій**. Вірніше, це не одна таблиця, а сім таблиць (своя таблиця для кожної з мелодій). Кожна таблиця позначена своєю окремою міткою (ті 11, ті 12 – ті 17). Значення кожної мітки – це адреса початку відповідної до мелодії. Кожне значення таблиці мелодій записується на згадку у вигляді одного байта. Тому всі рядки, крім останньої, для кожної таблиці мають парне число значень.

У **рядках 141,142** описана **таблиця початків усіх мелодій**. Початок цієї таблиці відзначене міткою tabm. Таблиця використовується для того, щоб програма могла знайти адресу початку потрібної мелодії по її номеру. У якості елементів масиву виступають подвоєні значення міток mel1, mel2 – mel7. Застосування подвоєних значень обумовлене необхідністю перекладу адрес із основної адресації в альтернативну. При трансляції програми замість міток на згадку будуть записані конкретні адреси.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 110</i>

Докладніше про основну й альтернативну адресацію читайте в [4-9].

Процедура обчислення адреси

Велика кількість таблиць у нашій програмі змушує подбати про процедуру обчислення адреси.

Однотипні обчислення зручно оформити у вигляді підпрограми. Ця підпрограма займає **рядки** 78–84. Виклик підпрограми проводиться по імені `addw`. Підпрограма одержує номер елемента таблиці й адреса її початку. Номер елемента передається в підпрограму за допомогою реєстру `YL`, а адреса – через реєстрову пару `Z`.

Використовуючи ці дані, підпрограма обчислює **адресу потрібного елемента**. Для цього вона спочатку подвоєє номер елемента (**рядок** 79). Потім доповнює отримане значення до шістнадцятирозрядну шляхом запису в `YN` нульового байта (**рядок** 80). І, нарешті, робить додавання двох шістнадцятирозрядних величин, що перебувають до цього моменту в реєстрових парах `Y` і `Z` (**рядка** 81, 82). Результат обчислень при цьому попадає, у реєстрову пару `Z`. Призначення команд `push i` і `pop` (**рядка** 78, 83), очевидно, уже пояснювати не потрібно.

Текст програми «крок за кроком»

Тепер розглянемо текст програми із самого початку. У рядках 3...10 розташований модуль опису змінних (робочих реєстрів). У рядках 13...31 розташовується модуль перевизначення векторів переривань, у рядках 32...41 – модуль команд ініціалізації.

Особливості програми

Процедура, розташована в **рядках** 45–52 програми, сканує клавіатуру й знаходить код першої з натиснутих кнопок. Найдений код перебуває в реєстрі `count`. Потім керування переходить

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 111</i>

до **рядка 53**. Із цього місця починається процедура вибору мелодії (**рядка 53–58**). Суть процедури – прочитати з таблиці `tabm` значення адреси початку цієї мелодії. Тобто прочитати елемент таблиці, номер якого дорівнює коду натиснутої клавіші.

Перш ніж прочитати елемент, необхідно **знайти його адресу**. Для обчислення адреси використовуємо підпрограму `addw`. Перед тем, як викликати підпрограму, підготуємо всі дані. Номер натиснутої клавіші поміщаємо в регістр `YL` (**рядок 53**). Адреса початку таблиці записуємо в регістрову пару `Z` (**рядка 54, 55**). І лише потім у **рядку 56** викликається підпрограма `addw`.

Після виходу з підпрограми в регістровій паре `Z` перебуває результат обчислень – адреса потрібного нам елемента таблиці `tabm`. Наступні дві команди (**рядка 57 і 58**) витягають той елемент (адреса початку мелодії) і поміщають його в регістрову пару `X`. Там ця адреса буде зберігатися увесь час, поки відтворюється саме ця мелодія.

Наступний етап – відтворення мелодії. Відтворенням мелодії займається процедура, розташована в **рядках 59–77**. Для послідовного відтворення нот нам знадобиться показчик поточної ноти. У якості показчика поточної ноти використовується регістрова пара `Z`. На самому початку процедури відтворення мелодії в регістрову пару `Z` міститься адреса початку мелодії їх регістрової пари `X` (**рядка 59, 60**).

Потім починається цикл відтворення (**рядка 61–77**). У цьому циклі програма витягає код ноти за адресою, на яку вказує наш показчик, виділяє з коду ноти код тону й код тривалості, відтворює ноту, а потім збільшує значення показчика на одиницю. Потім увесь цикл повторюється.

Цей процес відбувається доти, поки код чергової ноти не виявиться рівним 255 (мітка кінця мелодії). Прочитавши цей код, програма передає керування на **рядок 62**, де в регістр `Z` знову записується адреса початку мелодії. Відтворення мелодії почнеться

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 112</i>

спочатку. Цей процес повинен перерватися лише в одному випадку – при відпусканні керуючої кнопки.

Для перевірки стану кнопок у цикл відтворення мелодії включена спеціальна процедура (**рядка 61–63**). Процедура спрощено перевіряє стан відразу всіх кнопок. Вона зчитує вміст порту PD (**рядок 61**) і порівнює його з кодом 0x7F (**рядок 62**). Прочитане з порту значення може бути рівно 0x7F тільки в одному випадку – якщо всі кнопки відпущені. Якщо хоча б одна кнопка натиснута, то при читанні порту ми одержимо інше значення.

Перевіркою вищеописаної умови займається оператор **breq** у **рядку 63**. Якщо всі кнопки виявилися відпущені, цей оператор завершує цикл відтворення мелодії й передає керування на мітку **ml**, тобто на самий початок основного циклу програми. Там відбувається вимикання звуку, а потім нове сканування клавіатури.

Якщо хоча б одна кнопка виявиться натиснутою, то цикл відтворення звуку триває далі, і керування переходить до **рядка 64**, де відбувається добування коду ноти. Тому що адреса цієї ноти перебуває в реєстровій парі **Z** (покажчик поточної ноти), те для добування ноти просто використовується команда **lpm**.

У **рядку 65** відбувається перевірка ознаки кінця мелодії. Тільки що прочитаний код ноти рівняється з кодом 0x`ff`. Оператор **breq** у **рядку 66** передає керування по мітці **t4**, якщо мелодія дійсно закінчилася (умова виконується). Якщо код ноти не рівний 0x`ff`, переходу не відбувається, і керування переходить до **рядка 67**.

У **рядках 67–75** відбувається обробка коду ноти. Тобто з коду ноти виділяється код тону й код тривалості. Спочатку на код ноти накладається маска, яка залишає п'ять молодших розрядів, а три старші скидає (**рядок 67**). Під дією маски в реєстрі **temp** залишається код тону, який потім міститься в реєстрі **fnota** (**рядок 68**).

Тепер нам потрібно знайти **код тривалості ноти**. Для цього нам заново прийдє витягти код ноти з пам'яті програм. Тому що до

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 113

цього моменту ми не змінювали положення покажчика поточної ноти, то для добування немає ніяких перешкод. У **рядку 69** ми повторно витягаємо код ноти з пам'яті програм. Але цього разу значення покажчика збільшується. Тепер можна приступати до виділення коду тривалості. Як ви пам'ятаєте, тривалість кодується трьома молодшими бітами коду ноти. Для виділення цих бітів нам також потрібно використовувати **маску**. Але однієї маскою нам не обійтись. Нам потрібно не просто виділити три старші розряди, а зробити їхніми молодшими, як це показано на рис. 7.3.

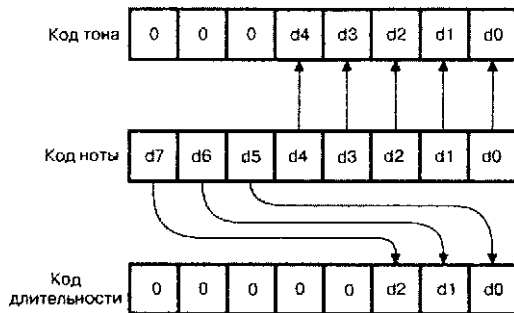


Рис. 6.3. Розкладання коду ноти

Процедура виділення коду тривалості займає рядки 70–74.

Спочатку програма робить багаторазове циклічне зрушення коду ноти доти, поки три старші розряди не стануть трьома молодшими. Для зрушення використовується команда `rol`. Тому що зрушення відбувається через гніздо ознаки переносу, то нам знадобиться чотири команди зрушення. Ці команди займають у програмі **рядка 70–73**.

Потім у **рядку 74** на отримане в результаті зрушень число накладається маска, яка виділяє три молодші біта, а п'ять старших скидає в нуль. Отриманий у такий спосіб код тривалості записується в регістр `dnote` (**рядок 75**).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 114</i>

Коли **код тону** й **код тривалості** визначені, проводиться виклик підпрограми відтворення ноти (**рядок 76**). Оператор `tjmr` у **рядку 77** передає керування на початок циклу відтворення мелодії, і цикл повторюється для наступної ноти.

Підпрограма відтворення ноти займає **рядки 85–110**. Вона виконує наступні дії:

- витягає з таблиці `tabkd` коефіцієнт розподілу, відповідний до коду ноти;
- програмує таймер і включає звук;
- потім витримує паузу й звук виключає.

Якщо код тону дорівнює нулю (потрібно відтворити паузу без звуку), добування коефіцієнта розподілу й включення звуку не виконується. Підпрограма відразу переходить до формування паузи.

Починається підпрограма відтворення ноти зі збереження всіх використовуваних реєстрів (**рядка 85–88**). Потім проводиться перевірка коду ноти на рівність нулю (**рядок 89**). Якщо код ноти дорівнює нулю, то оператор `breq` у **рядку 90** передає керування по мітці `ntl`, тобто до рядка, де відбувається виклик процедури формування затримки.

Якщо код ноти не дорівнює нулю, то програма приступає до добування коефіцієнта розподілу. Для обчислення адреси елемента таблиці `tabkd`, де перебуває цей коефіцієнт, знову використовується підпрограма `addw`.

Код тону міститься в реєстр `YL` (**рядок 91**), а адреса початку таблиці – у реєстрову пару `Z` (**рядка 92, 93**). Виклик підпрограми `addw` проводиться в **рядку 94**. У реєстровій парі `Z` підпрограма повертає адресу елемента таблиці, де перебуває потрібний нам коефіцієнт розподілу. У **рядках 95, 96** з таблиці дістається цей коефіцієнт. А в **рядках 97,98** він міститься в реєстр збігу таймера. У **рядках 99,100** включається звук.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 115</i>

У **рядку 104** викликається спеціальна підпрограма, призначена для формування затримки. Підпрограма називається `wait` і формує затримку зі змінною тривалістю. Тривалість затримки залежить від значення регістру `dnota`. По закінченню затримки звук вимикається (**рядка 102,103**).

На цьому можна було б закінчити процес відтворення ноти. Однак це ще не все. Для правильного звучання мелодії між двома сусідніми нотами необхідно забезпечити хоча б **невелику паузу**. Якщо такої паузи не буде, ноти будуть звучати разом. Це спотворить мелодію, особливо якщо підряд іде кілька нот з однаковим тоном. Формування паузи між нотами відбувається в **рядках 104,105**.

Допоміжна пауза формується за допомогою вже знайомій нам підпрограми затримки. У **рядку 104** коду паузи привласнюється нульове значення (вибирається сама мінімальна пауза). Потім у **рядку 105** викликається підпрограма `wait`. Після закінчення паузи залишається тільки відновити вміст усіх збережених регістрів зі стека (**рядка 106–109**) і вийти з підпрограми (**рядок 110**).

Підпрограма формування затримки

І останнє, що нам ще залишилося розглянути, – це підпрограма формування затримки. Текст підпрограми займає **рядки 111–135**. Як і будь-яка інша підпрограма, підпрограма `wait` на початку зберігає (**рядка 111–114**), а наприкінці – відновлює (**рядка 131–134**) усі використовувані регістри.

Розглянемо, як працює ця підпрограма. Спочатку визначається тривалість затримки. Для цього дістається відповідний елемент із таблиці `tabz`. Номер елемента відповідає коду затримки, що перебуває в регістрі `dnota`. Добування значення з таблиці проводиться вже знайомим нам образом. Команди, що реалізують обчислення адреси потрібного елемента таблиці, перебувають у **рядках 115–118**. Потім у

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	Екземпляр № 1	Арк 120 / 116

рядках 119 і 120 проводиться читання елемента таблиці. Прочитаний код затримки міститься в реєстрову пару Y .

Тепер **наше завдання**: *сформувати затримку, пропорційну вмісту реєстрової пари Y* . Тому що мікроконтролер Atiny2313 має тільки один шістнадцятирозрядний таймер, який уже зайнятий формуванням звуку, будемо формувати затримку програмним шляхом. Але в цьому випадку цикл формування затримки побудований небагато по-іншому.

Взагалі, способів побудови подібних підпрограм може бути нескінченна безліч. Усе залежить від винахідливості програміста. Використаний у даному прикладі спосіб більш зручний для формування затримки змінної тривалості, пропорційної заданому коефіцієнту. Головною **особливістю** нового способу є шістнадцятирозрядний параметр циклу.

Для зберігання цього параметра використовується реєстрова пара Z . Перед початком циклу затримки в неї записується нуль. Потім починається цикл, на кожному проході якого вміст реєстрової пари Z збільшується на одиницю. Після кожного такого збільшення проводиться порівняння нового значення Z із вмістом реєстрової пари Y .

Закінчується цикл тоді, коли вміст Z і вміст Y виявляться рівні. У результаті число, записане в реєстровій парі Y , буде визначати кількість проходів циклу. Тому й час затримки, формоване цим циклом, буде пропорційно константі затримки. Однак цей час буде занадто мало для одержання прийняттого темпу відтворення мелодій. Для того, щоб збільшити час до потрібної нам величини, усередину головного циклу затримки поміщений ще один цикл, що має фіксоване кількість проходів.

Описана вище процедура затримки займає **рядки 121–135**. У **рядках 121, 122** проводиться запис нульового знамення в реєстрову пару Z . Великий цикл затримки займає **рядки 123–130**. Малий внутрішній

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 117</i>

цикл займає **рядки 124–125**. Для зберігання параметра малого циклу використовується реєстр `loop`. У **рядку 123** у нього записується початкове значення. **Рядка 124,125** виконуються доти, поки вміст `loop` не виявиться рівним нулю.

У **рядку 126** вміст реєстрової пари `Z` збільшується на одиницю. У **рядках 127–130** проводиться порівняння вмісту двох реєстрових пар `Y` і `Z`. Порівняння проводиться побайтово. Спочатку рівняються молодші байти (**рядок 127**). Якщо вони не рівні, оператор умовного переходу в **рядку 128** передає керування на початок циклу.

Якщо молодші байти рівні, рівняються старші байти (**рядок 129**). Якщо старші байти неоднакові, оператор `brne` у **рядку 130** знову змушує цикл починатися з початку. І тільки коли обоє оператора порівняння дадуть позитивний результат (не викличуть переходу), цикл закінчується, і підпрограма формування затримки переходить до завершальної фази (до **рядків 131–135**).

6.4. Порядок виконання роботи

1. Створити новий проект в середовищі PROTEUS. Скласти схему музичної шкатулки згідно рис. 7.1.

2. Створити новий проект в AVR Studio, вибрати New Project, далі вибрати як налагоджувальну платформу Atmel AVR Assembler, AVR Simulator і мікросхему ATtiny2313.

3. Для мікроконтролера ATtiny2313 написати програму музичної шкатулки на дві мелодії. Вибрана мелодія повинна грати при натисканні кнопки. Варіант мелодій задає викладач.

Для створення програми користуватися інформацією з теоретичних відомостей та прикладом 7.1.

4. Виконати відладку програму в AVR Studio в покроковому режимі за допомогою миші або кнопки F11.

5. Відкрити створений в PROTEUS проект і підключити написану програму до мікропроцесора. Запустити симуляцію, перевірити працездатність схеми.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 118</i>

6. На основі першої програми створити музичну шкатулку у якій мелодії будуть звучати в іншій тональності – на октаву нижче.

7. Відкрити створений в PROTEUS проект і підключити написану програму до мікропроцесора. Запустити симуляцію, перевірити працездатність схеми.

8. Написати звіт. Зробити висновки.

6.5. Зміст звіту

Звіт повинен містити:

1. Тексти написаних програм.
2. Перелік використаних команд з поясненням їх призначення.
3. Зробити висновки: про об'єм виконаної роботи, чи досягнена мета роботи, чи практичні результати співпали з теоретичними.

6.6. Контрольні тестові питання

1. Який склад схеми музичної шкатулки?
2. Поясніть алгоритм роботи музичної шкатулки з використанням МК?
3. Який принцип програмування нот?
4. Яка підпрограма формування затримки може бути використана?
5. Які дії необхідно провести, щоб проемувлювати в PROTEUS роботу мікроконтролера?
6. Які нові команди використано в розробленій програмі?
7. Які регістри та інші складові архітектури МК задіяні в роботі розробленої програми?
8. Як задати необхідний час симуляції?

Література

1. Схемотехніка електронних систем: У 3-х кн. – Кн. 2. Цифрова схемотехніка : підручник / В.І. Бойко, А.М. Гуржій, В.Я. Жуйко та ін. – 2-ге вид., доп. та перероб. – К. : Вища школа, 2004. – 423 с.
2. Мікропроцесорна техніка. Підручник / Ю.І. Якименко, Т.О. Терещенко, Є.І. Сокол, В.Я. Жуйкою, Ю.С. Петергеря ; за

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.02/2/172.00.1/М /ВК2.4-2021
	<i>Екземпляр № 1</i>	<i>Арк 120 / 119</i>

ред. Т.О. Терещенка, 2-е видання, перероблене та доповнене. – К. : ІВЦ. Вид-во «Політехніка», «Кондор», 2008. – 594 с.

3. Схемотехніка електронних систем: У 3-х кн. – Кн. 3. Мікропроцесори та мікроконтролери : підручник / В.І. Бойко, А.М. Гуржій, В.Я. Жуйко та ін. – 2-ге вид., доп. та перероб. – К. : Вища школа, 2004. – 399 с.
4. Бабак В.П. та ін. Обробка сигналів: Підручник /В.П. Бабак, В.С. Хандецький, Е. Шрюфер. – К: Либідь, 1996. -392с.
5. Proakis J.G. (2006). Digital Signal Processing, Principles, Algorithms, and Applications. 4-th Edition. – New Jersey: Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1077.