

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни
“Інтерфейси та пристрої зв’язку з об’єктом”

для студентів спеціальності 8.05010203 “Спеціалізовані
комп’ютерні системи” всіх форм навчання

Частина 2

2015

Методичні вказівки до лабораторних робіт з дисципліни „Інтерфейси та пристрої зв'язку з об'єктом” для студентів спеціальності 8.05010203 “Спеціалізовані комп'ютерні системи” всіх форм навчання. Частина 2 / Сост.: А.В. Сирота, А.В. Тіменко. – Запоріжжя: ЗНТУ, 2015. – 46 с.

Рекомендовано НМК як методичні вказівки до лабораторних робіт з дисципліни „Інтерфейси та пристрої зв'язку з об'єктом” усіх форм навчання.

Укладачі:	А.В. Сирота, ст. викладач, А.В. Тіменко, асистент.
Рецензент:	Р.К. Кудерметов, к.т.н., доцент
Відповідальний за випуск:	А.В. Сирота, ст. викладач.

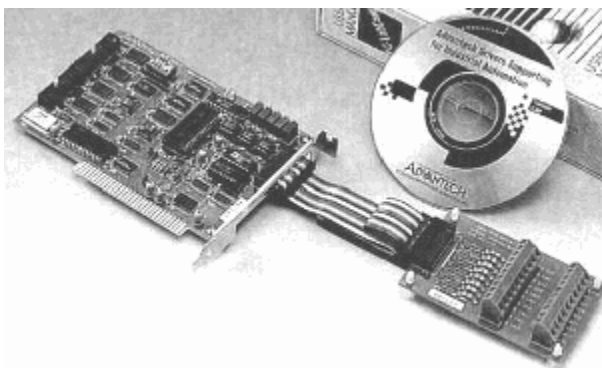
Затверджено на засіданні
кафедри „Комп'ютерні системи і мережі”
Протокол № 6
від 02.02.2015р.

ЗМІСТ

ВСТУП	4
1 ВИВЧЕННЯ УНІВЕРСАЛЬНОЇ ПЛАТИ ЗБОРУ ДАНИХ PCL-711.....	5
1.1 Основні характеристики	5
1.2 Призначення перемикачів.....	6
1.3 Зовнішні з'єднувачі	7
1.4 Регістрова модель плати	9
1.5 Регістри даних АЦП.....	10
1.6 Регістр керування посиленням сигналу.....	11
1.7 Регістр вибору каналу	11
1.8 Регістр керування режимом і перериваннями	12
1.9 Регістр стану переривання	14
1.10 Регістр програмного запуску АЦП	14
1.11 Регістри ЦАП	14
1.12 Регістри дискретного введення.....	15
1.13 Регістри дискретного виведення.....	15
1.14 Опис лабораторного стенду	16
1.15 Хід роботи	16
1.15.1 Створення моделі	17
1.15.2 Робота з моделлю	21
1.16 Завдання на роботу.....	22
1.17 Зміст звіту.....	22
1.18 Контрольні запитання	23
2 РЕЖИМИ ПРОГРАМНОГО ОБМІНУ ДАНИМИ.....	24
2.1 Реалізація введення аналогових даних плати PCL-711	24
2.2 Навчальна програма PCL711.CPP.....	25
2.3 Загальна організація PCL711.CPP.....	26
2.4 Організація програмного запуску АЦП.....	28
2.5 Організація запуску АЦП зовнішнім сигналом.....	30
2.6 Регістри програмованого таймера Intel 8253.....	32
2.7 Регістр керування лічильником	33
2.8 Режими роботи таймера/лічильника.....	34
2.9 Програмування мікросхеми Intel 8253.....	35
2.10 Організація запуску АЦП сигналом таймера.....	36
2.11 Програмування системного таймеру ПК	38
2.12 Достовірність результатів спостереження	41
2.13 Завдання на роботу.....	44
2.14 Зміст звіту.....	45
2.15 Контрольні запитання	45
ЛІТЕРАТУРА	46

ВСТУП

Плата **PC-MultiLab PCL-711S** (далі просто PCL-711) - зручна в роботі, недорога, сумісна з IBM PC/XT/AT багатофункціональна плата для збору даних. Технічні можливості PCL-711 роблять її популярним рішенням в широкому діапазоні промислових і дослідницьких задач. Можливі напрями застосування плати PCL-711: збір даних, управління процесами, автоматичний контроль і промислова автоматизація.



Зовнішній вигляд плати PCL-711

Основні технічні характеристики плати PCL-711:

- аналого-цифрове перетворення (АЦП): 12 біт;
- кількість каналів АЦП: 8 аналогових входів із загальним дротом;
- діапазон вхідної напруги встановлюється програмно: $\pm 5V \pm 2.5V \pm 1.25V \pm 0.625V \pm 0.3125V$;
- режим запуску АЦП: програмний, від зовнішнього сигналу, від вбудованої схеми запуску;
- рівень переривання для передачі даних АЦП, що програмно налаштовується;
- один 12-розрядний канал дискретного виведення, що перемикається.
- діапазон напруги цифро-аналогового перетворювача (ЦАП): 0..5V або 0..10V;
- вбудовані 16-розрядні канали дискретного уведення-виведення.

1 ВИВЧЕННЯ УНІВЕРСАЛЬНОЇ ПЛАТИ ЗБОРУ ДАНИХ PCL-711

Лабораторна робота № 3

Мета роботи. Вивчити регістрову модель і характеристики універсальної плати аналогового і цифрового уведення-виведення PCL-711. Навчитися використовувати пакет прикладних програм (ППП) Windows Real-Time Target системи моделювання MATLAB для збору даних за допомогою плати PCL-711.

1.1 Основні характеристики

Аналого-цифрове перетворення:

- кількість каналів 8 із загальним дротом;
- дозвіл АЦП;
- діапазон вхідної напруги встановлюється програмно: $\pm 5\text{В}$ $\pm 2.5\text{В}$ $\pm 1.25\text{В}$ $\pm 0.625\text{В}$ і $\pm 0.3125\text{В}$;
- мікросхема АЦП AD574 (12 біт) або еквівалентний;
- час перетворення: 25мкс максимум;
- межа основної погрішності вимірювання: 0.015% значення ± 1 MSB;
- коефіцієнт посилення: $\times 1$, $\times 2$, $\times 4$, $\times 8$ і $\times 16$ (встановлюється програмно);
- режим запуску: програмний, від зовнішнього сигналу, від вбудованої схеми запуску;
- передача даних: командою програми або за перериванням;
- допустиме перевантаження на вході: постійне ± 30 В максимум.

Цифро-аналогове перетворення:

- кількість каналів: 1 канал;
- діапазон вихідної напруги: 0..5В або 0..10В;
- час встановлення: 30мкс;
- опорна напруга: від внутрішнього джерела, -5В і -10В ($\pm 0.05\text{В}$);
- мікросхема ЦАП PM7548 (12 біт) або еквівалентний;
- нелінійність: ± 0.5 MSB;
- максимальне навантаження: $\pm 5\text{мА}$ максимум.

Максимальна частота дискретизації, що використовується для збору даних за допомогою плати PCL-711 - 25 кГц. Проте, швидкість передачі даних від плати PCL-711 в комп'ютер обмежена. На це впливають архітектура обчислювальної системи і програмне забезпечення, що використовується. Швидкість передачі може змінюватися залежно від мови програмування, що ви-

користовується, ефективності коду, і т. ін. Наприклад, якщо використовується режим передачі даних від PCL-711 за перериванням під керуванням операційної системи Windows, то дані можна отримати тільки у тому випадку, якщо частота надходження відліків більше частоти виконання процедури обробки переривання PCL-711.

1.2 Призначення перемикачів

Конструкція плати PCL-711 передбачає прості засоби конфігурування. На платі розташована ланка DIP-перемикачів (SW1) і один комплект для установлення перемички (JP1). Вони використовуються для вибору базової адреси порту уведення-виведення і діапазону вихідної напруги ЦАП відповідно.

Керування більшістю периферійних пристроїв і плат розширення комп'ютера виконується за допомогою портів уведення-виведення. Периферійні пристрої повинні використовувати власні, окремі простори портів уведення-виведення, що не перекриваються. Плата PCL-711 використовує 16 послідовно розташованих адрес в просторі портів уведення - виведення обчислювальної системи.

Базова адреса порту уведення - виведення вибирається за допомогою 6-позиційного DIP-перемикача SW1. Може бути вибрана адреса в діапазоні від 000h до 3F0h. Заводська установка – 220h. В таблиці 1.1 наведені приклади настройки базової адреси простору уведення-виведення плати PCL-711.

Діапазон вихідної напруги ЦАП плати PCL-711 залежить від опорної напруги, яка задається перемичкою JP1. Опорна напруга може приймати два значення: -5В або -10В.

Таблиця 1.1 – Настройка базової адреси простору уведення-виведення

Діапазон адрес портів	Перемикачі SW1					
	1	2	3	4	5	6
	A9	A8	A7	A6	A5	A4
000h-00Fh	0	0	0	0	0	0
100h-10Fh	0	1	0	0	0	0
-	-	-	-	-	-	-
200h-20Fh	1	0	0	0	0	0
210h-21Fh	1	0	0	0	0	1
220h-22Fh	1	0	0	0	1	0
-	-	-	-	-	-	-
300h-30Fh	1	1	0	0	0	0
-	-	-	-	-	-	-
3F0h-3FFh	1	1	1	1	1	1

Тут: 0 = ON, 1 = OFF. Перемикачі 1..6 відповідають розрядам A9..A4 шини адреси комп'ютера.

Якщо потрібен діапазон вихідних напруг ЦАП 0..5В, то перемикачка встановлюється, як показано на рис. 1.1,а. Це заводська настройка, що використовується за умовчанням. Якщо потрібен діапазон 0..10В, то перемикачка встановлюється, як показано на рис. 1.1,б.

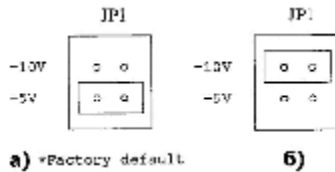


Рисунок 1.1 – Вибір діапазону вихідної напруги ЦАП

1.3 Зовнішні з'єднувачі

Плата PCL-711 обладнана трьома зовнішніми 20-контактними з'єднувачами. Два з'єднувачі, CN3 і CN4, використовуються для дискретного введення (CN4) і дискретного виведення (CN3). З'єднувач CN1 використовується для аналогового введення-виведення.

Усі з'єднувачі мають однаковий зовнішній вигляд, що дозволяє використовувати уніфікований стрічковий кабель.

На рис. 1.2 показані призначення контактів зовнішніх з'єднувачів, а в таблиці 1.2 - скорочення, що використовуються.

Analog I/O (CN1)				Digital Output (CN3)				Digital Input (CN4)			
A/D 0	1	2	AGND	D/O 0	1	2	D/O 1	D/I 0	1	2	D/I 1
A/D 1	3	4	AGND	D/O 2	3	4	D/O 3	D/I 2	3	4	D/I 3
A/D 2	5	6	AGND	D/O 4	5	6	D/O 5	D/I 4	5	6	D/I 5
A/D 3	7	8	AGND	D/O 6	7	8	D/O 7	D/I 6	7	8	D/I 7
A/D 4	9	10	AGND	D/O 8	9	10	D/O 9	D/I 8	9	10	D/I 9
A/D 5	11	12	AGND	D/O 10	11	12	D/O 11	D/I 10	11	12	D/I 11
A/D 6	13	14	AGND	D/O 12	13	14	D/O 13	D/I 12	13	14	D/I 13
A/D 7	15	16	AGND	D/O 14	15	16	D/O 15	D/I 14	15	16	D/I 15
D/A	17	18	AGND	DGND	17	18	DGND	DGND	17	18	DGND
AGND	19	20	AGND	+5V	19	20	+12V	+5V	19	20	STROB

Рисунок 1.2 – Внешние соединители платы PCL-711

Підключення для уведення аналогового сигналу. Плата PCL-711 обслуговує вісім асиметричних аналогових каналів. Асиметричне аналогове підключення використовує для кожного терміналу окремий сигнальний дріт, і один загальний дріт, один для всіх аналогових каналів (сполучений з AGND). На рис.1.3 показане асиметричне підключення, що використовує загальний дріт.

Таблиця 1.2

Скорочення	Призначення	
A/D	Analog input	Аналоговий вхід
AGND	Analog ground	Аналогова земля
D/A	Analog output	Аналоговий вихід
D/O	Digital output	Дискретний вихід
D/I	Digital input	Дискретний вхід
DGND	Digital ground	Цифрова земля
VREF	Voltage reference	Опорна напруга
STROBE	External signal STROBE	Фіксація даних

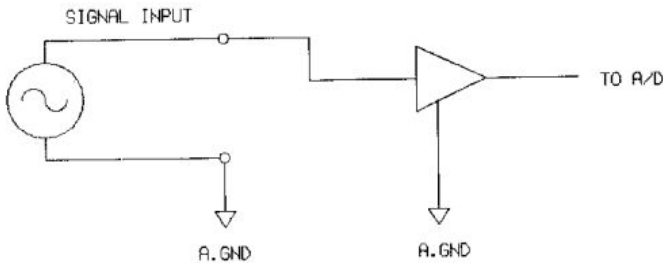


Рисунок 1.3 - Асиметричне підключення аналогового входу

Плата PCL-711 не призначена для диференціального підключення аналогових сигналів.

Підключення для виведення аналогового сигналу. Плата PCL-711 обладнана джерелом опорної напруги -5В і -10В, за допомогою якого можна встановлювати діапазон вихідної напруги ЦАП. Є один канал виведення аналогового сигналу ЦАП (контакти 17 і 19 з'єднувача CN1). На рис.1.4 показано типове підключення для аналогового виведення і ЦАП.

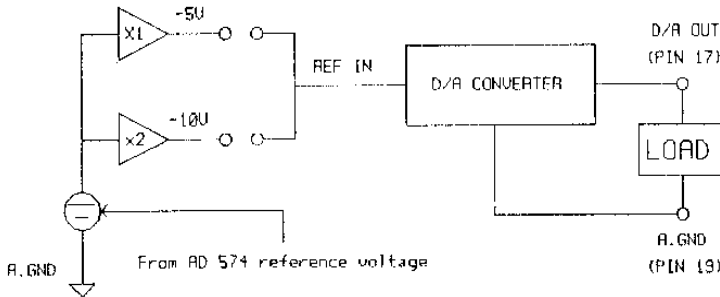


Рисунок 1.4 – Підключення для виведення аналогового сигналу

Підключення для уведення-виведення дискретних сигналів. Плата PCL-711 обслуговує 16 дискретних каналів уведення-виведення. Вони можуть використовуватися, наприклад, для операцій типу "таймер/лічильник". Рівень напруги всіх цифрових каналів сумісний з TTL.

Для організації прийому/передачі дискретних сигналів і зв'язку із зовнішніми пристроями, сумісними з TTL, використовується підключення, наведене на рис.1.5.

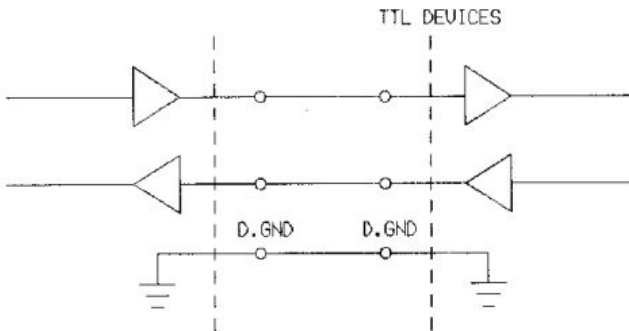


Рисунок 1.5 – Підключення для дискретного уведення-виведення

1.4 Регістрова модель плати

В таблиці 1.3 показаний простір портів уведення-виведення плати PCL-711. Для керування платою використовуються 16 регістрів, що послідовно адресуються, кожний з яких призначений для виконання тієї або іншої функції. Далі кожний регістр буде розглянутий детально.

Таблиця 1.3 – Простір портів введення-виведення

Адреса порту	Читання	Запис
BASE+0	Лічильник 0	Лічильник 0
BASE+1	Лічильник 1	Лічильник 1
BASE+2	Лічильник 2	Лічильник 2
BASE+3	не доступний	Управління лічильником
BASE+4	Дані АЦП, м.байт	Дані ЦАП, м.байт
BASE+5	Дані АЦП, ст.байт	Дані ЦАП, ст.байт
BASE+6	Дискретний вхід, м.байт	не доступний
BASE+7	Дискретний вхід, ст. байт	не доступний
BASE+8	не доступний	Скидання стану переривання
BASE+9	не доступний	Управління посиленням
BASE+10	не доступний	Вибір каналу АЦП
BASE+11	не доступний	Управління режимом і перериванням
BASE+12	не доступний	Програмний запуск АЦП
BASE+13	не доступний	Дискретний вихід, м.байт
BASE+14	не доступний	Дискретний вихід, ст.байт
BASE+15	не доступний	не доступний

1.5 Регістри даних АЦП

Для зберігання результату АЦП використовуються два регістри даних, що розташовані за адресами BASE+4 і BASE+5. Молодший байт – в регістрі BASE+4, старший байт – в регістрі BASE+5.

BASE+4: Молодший байт даних АЦП (Читання)

D7	D6	D5	D4	D3	D2	D1	D0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

BASE+5: Старший байт даних АЦП (Читання)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	DRDY	AD11	AD10	AD9	AD8

Тут:

AD0..AD11 – інформаційні розряди АЦП. AD0 – наймолодший розряд (LSB), AD11 – самий старший розряд (MSB).

DRDY – прапор готовності даних. Доки виконується АЦП прапор DRDY=1. Коли АЦП завершується DRDY=0. Встановлюється в 1 після читання молодшого байта даних з регістра BASE+4.

1.6 Регістр керування посиленням сигналу

Регістр BASE+9 використовується для настройки коефіцієнта посилення сигналу АЦП плати PCL-711. Передбачено п'ять можливих значень: x1, x2, x4, x8, і x16. В таблиці 1.3 приведені коди настройки коефіцієнта посилення.

BASE+9: Настройка посилення сигналу (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	G2	G1	G0

Таблиця 1.4 – Коди настройки коефіцієнта посилення

G2	G1	G0	Посилення
0	0	0	x1
0	0	1	x2
0	1	0	x4
0	1	1	x8
1	0	0	x16

1.7 Регістр вибору каналу

Конструкція плати PCL-711 передбачає перемикання до 8 каналів аналогового введення. Для того, щоб вибрати потрібний канал, перед виконанням АЦП користувач повинен записати відповідний код в регістр BASE+10. Призначення розрядів цього регістра вказано нижче.

BASE+10: Вибір каналу аналогового введення (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	C2	C1	C0

Таблиця 1.5 – Коди настройки каналу аналогового введення

C2	C1	C0	Канал
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

1.8 Регістр керування режимом і перериваннями

Аналого-цифрове перетворення може бути ініційоване одним з трьох способів.

1. **Командою програми.** Якщо записувати будь-які дані за адресою BASE+12, то автоматично генеруватиметься сигнал запуску АЦП на платі PCL-711.
2. **Зовнішнім сигналом.** Зовнішній сигнал запуску АЦП треба подавати на вхід D/I 0 (контакт 1 з'єднувачу CN4). Аналого-цифрове перетворення починається по передньому фронту цього сигналу, коли його логічний стан змінюється з "0" на "1".
3. **Вбудованою системою запуску.** Для завдання часових інтервалів плата PCL-711 обладнана мікросхемою програмованого інтервального таймера-лічильника Intel 8253. Тактові імпульси можуть генеруватися в діапазоні від 0.5МГц і 35 хв./імпульс.

Також передбачено два способи організації передачі даних, що отримані в результаті АЦП.

1. **Програмний.** Такий спосіб передачі даних передбачає програмний опит стану прапора готовності даних DRDY (біт D4 регістра BASE+5). Спочатку виконується запуск АЦП, а потім прикладна програма повинна постійно перевіряти значення біта DRDY, поки він не встановиться в 0 (ознака закінчення АЦП). Після цього можна прочитати дані, що зберігаються в регістрах BASE+4 і BASE+5 (результат роботи АЦП).
2. **За перериванням.** Використовування переривання дозволяє одержувати дані з PCL-711 у фоновому режимі. В цьому випадку по закінченню кожної операції АЦП генеруватиметься запит за перериванням (Interrupt Request, **IRQ**). Його обслуговуванням повинна займатися відповідна процедура обробки переривання (Interrupt Service Routine, **ISR**), основне призначення якої – прочитати дані, що зберігаються в регістрах BASE+4 і BASE+5, і передати для обробки основній програмі.

Регістр BASE+11 використовується для настройки режиму роботи PCL-711 і рівня IRQ. Формат регістра має наступний вигляд.

BASE+11: Настройка режиму і рівня переривання (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
-	I2	I1	I0	-	S2	S1	S0

Таблиця 1.6 – Коды режимів роботи плати

S2	S1	S0	Режим роботи
0	0	0	Запуск по команді програми і програмний спосіб передачі даних

0	0	1	
0	1	0	Запуск від зовнішнього сигналу і програмний спосіб передачі
0	1	1	Запуск від зовнішнього сигналу і передача по перериванню
1	0	0	Запуск від вбудованої системи і програмний спосіб передачі
1	0	1	Зарезервовано
1	1	0	Запуск від вбудованої системи і передача по перериванню
1	1	1	Зарезервовано

Таблиця 1.7 – Коды настройки рівня переривання

I2	I1	I0	Переривання
0	0	0	IRQ2
0	0	1	-
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

1.9 Регістр стану переривання

В режимі передачі даних за перериванням після кожного аналого-цифрового перетворення встановлюється прапор, що повідомляє про це. Для того, щоб продовжити інформаційний обмін, в регістр BASE+8 необхідно записати будь-яку двійкову послідовність. В результаті прапор буде скинений, і плата PCL-711 буде готова передавати нову порцію даних.

BASE+8: Скидання стану переривання (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	-	-

1.10 Регістр програмного запуску АЦП

Запис будь-яких даних в регістр BASE+12 призводить до того, що генерується сигнал запуску АЦП плати PCL-711.

BASE+12: Скидання стану переривання (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	-	-

1.11 Регістри ЦАП

PCL-711 має один канал дискретного виведення. Молодший байт зберігається в регістрі BASE+4, старший байт – в регістрі BASE+5.

Для більшої надійності, для уведення-виведення дискретних даних використовується технологія **подвійної буферизації**, суть якої полягає в тому, що інформаційний обмін здійснюється в два етапи. Спочатку обробляється молодший байт (регістр BASE+4), а потім старший байт (регістр BASE+5). Таким чином, інформація, що поступає в ЦАП, не міняється, поки не буде виконано звернення до регістра BASE+5.

BASE+4: Молодший байт даних ЦАП (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

BASE+5: Старший байт даних ЦАП (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	DA11	DA10	DA9	DA8

Тут:

DA0..DA11: дані ЦАП. DA0 – наймолодший значущий розряд, DA11 – самий старший значущий розряд.

1.12 Регістри дискретного уведення

Плата PCL-711 використовує 16-розрядний канал дискретного уведення інформації. В просторі уведення-виведення відповідні регістри мають адреси BASE+6 і BASE+7.

BASE+6: Молодший байт каналу дискретного уведення (Читання)

D7	D6	D5	D4	D3	D2	D1	D0
DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

BASE+7: Старший байт каналу дискретного уведення (Читання)

D7	D6	D5	D4	D3	D2	D1	D0
DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8

1.13 Регістри дискретного виведення

Плата PCL-711 використовує 16-розрядний канал дискретного виведення інформації. В просторі уведення-виведення відповідні регістри розташовані за адресами BASE+13 і BASE+14.

BASE+13: Молодший байт каналу дискретного виведення (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

BASE+14: Старший байт каналу дискретного виведення (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8

1.14 Опис лабораторного стенду

Лабораторний стенд організований на базі персонального комп'ютера (ПК) і складається з плати PCL-711, плати генератора прямокутних імпульсів (ГПІ) і двох 20-контактних стрічкових кабелів. Збирання стенду виконується за вимкненої напруги живлення в наступному порядку.

1. Фізично відключити електроживлення системного блоку ПК. Зняти захисний кожух системного блоку.
2. Встановити плату PCL-711 у вільний з'єднувач ISA материнської плати ПК. Якщо необхідно, за допомогою перемикачів SW1 встановити початкову адресу простору портів введення-виведення плати PCL-711 (див. табл.1.1).
3. Встановити плату ГПІ в системний блок.
4. Вільний з'єднувач блоку живлення ПК з'єднати з з'єднувачем електроживлення ГПІ.
5. З'єднувачі першого 20-контактного сполучного кабелю встановити в з'єднувачі CN3 і CN4 плати PCL-711. За рахунок цього сигнали дискретного виведення плати подаються в канал дискретного введення.
6. Перший вільний з'єднувач другого 20-контактного сполучного кабелю встановити в з'єднувач CN1 плати PCL-711, а другий – в зовнішній з'єднувач плати ГПІ. За рахунок цього сигнали з виходу плати ГПІ поступають в канал аналогового введення плати PCL-711.
7. Встановити захисний кожух системного блоку ПК. Включити електроживлення.
8. За нормальної роботи повинні миготіти світлодіоди плати ГПІ.

1.15 Хід роботи

Під час використання стандартних засобів збирання даних, таких як плата PCL-711, іноді виникає необхідність отримати швидкий результат, не відволікаючись на питання програмування.

Під час виконання практичної частини лабораторної роботи вивчаються наступні питання.

1. Методика побудови моделі засобами середовища моделювання (СМ) Simulink системи комп'ютерної математики (СКМ) MATLAB.
2. Методика використання ППП Real-Time Windows Target CM Simulink.
3. Методика автоматичного складання програм мовою C/C++ для системи розробки програм (СПП) Microsoft Visual C засобами ППП Real-Time Workshop СКМ MATLAB.

1.15.1 Створення моделі

- 1.15.1.1 Запустіть СКМ MATLAB. У вікні команд наберіть команду **simulink**. З'явиться нове вікно, що містить бібліотеку компонентів CM Simulink.
- 1.15.1.2 Виберіть команду **File** → **New** → **Model**. З'явиться нове вікно – документ моделі CM Simulink.
- 1.15.1.3 Збережіть файл моделі на диску ПК.
- 1.15.1.4 Серед компонентів ППП Real-Time Windows Target CM Simulink знайдіть об'єкт **Analog Input** і перетягніть його на документ моделі.

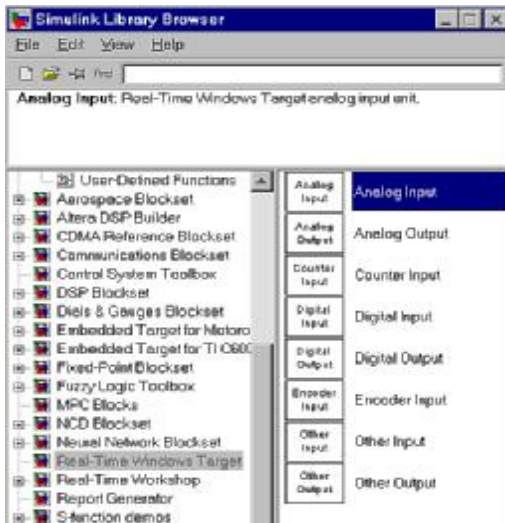


Рисунок 1.6 – Бібліотека компонентів Simulink

- 1.15.1.5 Серед компонентів ППП **Simulink** → **Sinks** знайдіть об'єкт **Scope** і перетягніть його на документ моделі.
- 1.15.1.6 Серед компонентів ППП **Simulink** → **Math Operations** знайдіть об'єкт **Gain** і перетягніть його на документ моделі.
- 1.15.1.7 З'єднайте всі об'єкти документа моделі, створивши схему, що наведена на рис.1.7.

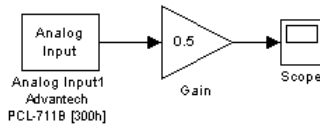


Рисунок 1.7 – Схема моделі

- 1.15.1.8 Виконайте настройку блоку **Analog Input**. Для цього двічі клацніть на ньому. З'явиться вікно **Block Parameters: Analog Input** (рис.1.8).

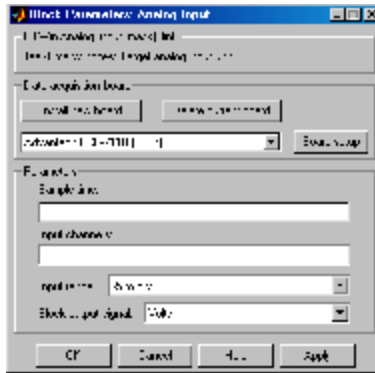


Рисунок 1.8 – Налаштування блоку Analog Input

- 1.15.1.9 У вікні **Block Parameters: Analog Input** клацніть кнопку **Install new board**. З'явиться список стандартних пристроїв збирання даних. Виберіть об'єкт **Advantech** → **PCL711B**.
- 1.15.1.10 У вікні **Block Parameters: Analog Input** клацніть кнопку **Board setup**. З'явиться вікно налаштування початкової адреси простору портів введення-виведення плати PCL711.



Рисунок 1.9 – Налаштування початкової адреси плати PCL-711

- 1.15.1.11 Для того, щоб переконаватися в тому, що програмний інтерфейс ППП Real-Time Windows Target знайшов плату PCL711, клацніть кнопку

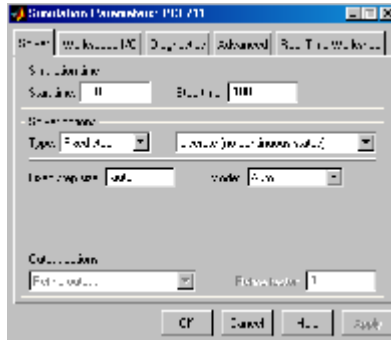


Рисунок 1.11 – Параметри моделювання

1.15.1.17 У вікні **Simulation parameters** виберіть вкладку **Real-Time Workshop**. Клацніть кнопку **Browse**. З'явиться вікно **System Target File Browser** (рис.1.12).

Таблиця 1.13 – Параметри моделювання

Параметр	Значення	Пояснення
Start time	0.0	Час початку роботи
Stop time	300.0	Час закінчення роботи
Type	Fixed step	Тип моделі
Fixed step size	auto	Розмір кроку моделювання
Mode	Auto	Режим

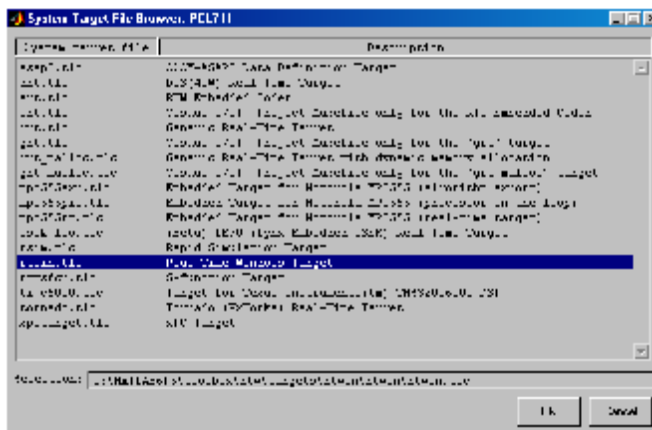


Рисунок 1.12 – Вибір Real-Time Windows Target

1.15.1.18 Виберіть файл **rtwin.tlc** – **Real-Time Windows Target**. Спочатку клацніть кнопку **Apply**, потім – **OK**. На рис.1.13 показаний результат настроювання вкладки **Real-Time Workshop** вікна **Simulation parameters**.

1.15.1.19 Модель створена. Збережіть файл моделі на диску ПК.

1.15.2 Робота з моделлю

На рис.1.14 зображена панель інструментів документу моделі.

1.15.2.1 В комбінованому списку виберіть тип акселератора Simulink – **External**.

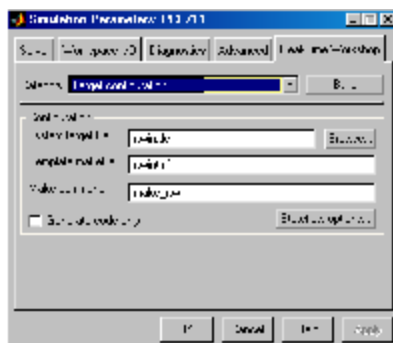


Рисунок 1.13 – Настроювання Real-Time Workshop



Рисунок 1.14 – Панель інструментів моделі

1.15.2.2 Клацніть кнопку **Save model** (зберегти документ моделі).

1.15.2.3 Клацніть кнопку **Update diagram** (модифікувати схему моделі).

1.15.2.4 Клацніть кнопку **Build all** (побудувати модель). В командному вікні СКМ MATLAB з'являться повідомлення про хід створення файлів мовою C, що належать моделі. Вони будуть автоматично транслюватися СРП Visual C. В робочому каталозі на диску з'являться відповідні об'єктні файли.

1.15.2.5 Для того, щоб за допомогою створеної моделі можна було збирати дані, треба клацнути на кнопці **Connect to target** (підключитися до цілі). Модель перейшла в режим очікування команд користувача.

- 1.15.2.6 Кладнувши на кнопки **Start real-time code**, запустить модель.
- 1.15.2.7 Двічі клацніть на об'єкті **Scope**, що розміщений на документі моделі. З'явиться вікно, де відображається часова діаграма сигналу, що надходить від ГПП на перший канал АЦП плати PCL-711 (рис.1.15).

1.16 Завдання на роботу

1. Вивчити технічні характеристики і реєстрову модель плати PCL-711.
2. Виконати практичну частину роботи.
3. Приєднайте додаткові об'єкти до моделі і виконайте спостереження аналогових даних, що надходять каналом 2 АЦП плати PCL-711.
4. Прогляньте вміст файлів мовою C, що були автоматично створені СКМ MATLAB і збережені в директорії документу моделі. Зробіть висновки.

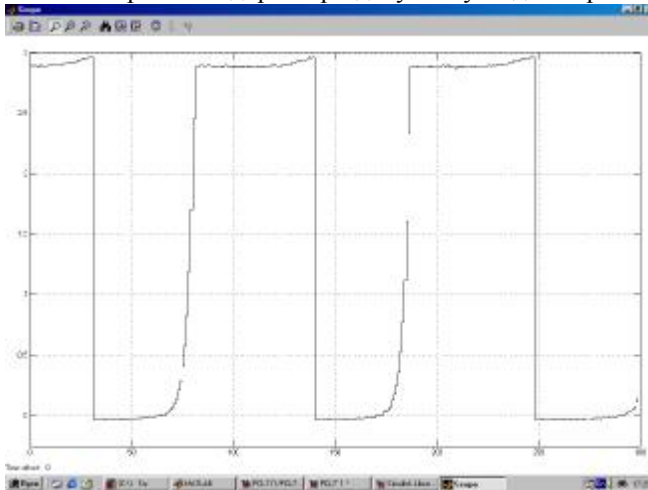


Рисунок 1.15 – Результат спостережень

1.17 Зміст звіту

1. Титульний лист. Мета роботи.
2. Короткий конспект – опис реєстрової моделі плати PCL-711.
3. Останній варіант схеми моделі, що дозволяє одночасно збирати данні з двох каналів АЦП.
4. Висновки.

1.18 Контрольні запитання

1. Призначення і функціонально-структурна організація плати PCL-711.
2. Технічні характеристики каналу аналогового уведення, типова схема підключення.
3. Технічні характеристики каналу аналогового виведення, типова схема підключення.
4. Регістрова модель плати PCL-711. Які регістри по різному використовуються в режимі читання і режимі запису даних?
5. Три способи запуску АЦП плати PCL-711.
6. Два способи організації передачі даних плати PCL-711. Який ще третій спосіб організації даних від ПЗО існує, але в платі PCL-711 не реалізований?
7. Порядок створення моделі в СМ Simulink СКМ MATLAB.

2 РЕЖИМИ ПРОГРАМНОГО ОБМІНУ ДАНИМИ

Лабораторна Робота № 4

Мета роботи. Вивчити принципи організації програмного способу передачі даних універсальної плати аналогового і цифрового введення-виведення PCL-711. Навчитися програмувати таймер Intel 8253 в режимі генератора частоти для вмикання АЦП.

2.1 Реалізація введення аналогових даних плати PCL-711

На рис.2.1 зображена структурно-функціональна організація плати PCL-711. В таблиці наведені пояснення до цієї схеми.

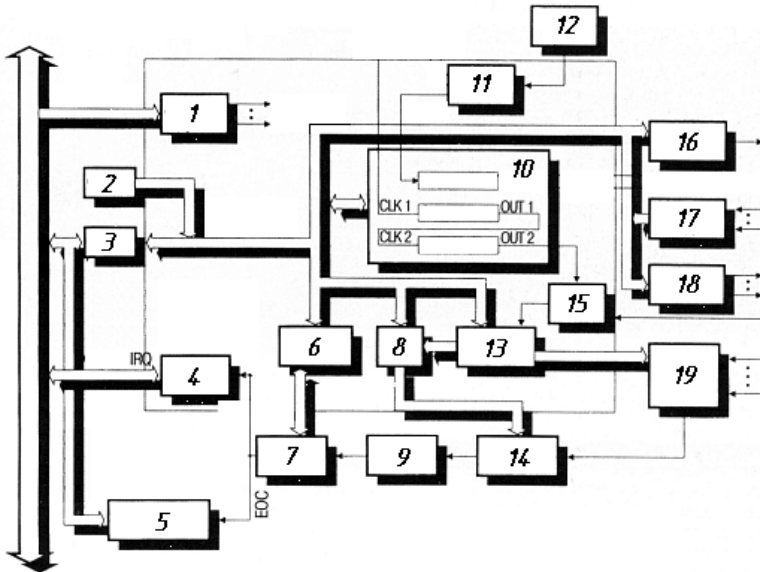


Рисунок 2.1 – Структурно-функціональна організація плати PCL-711

В режимі збирання даних, що, в загальному випадку, надходять на 8 каналів аналогового введення виконується така послідовність дій.

1. Схема сканування каналів (13) керує аналоговим мультіплексором (19), який вмикає потрібний канал аналогового введення.

2. Під впливом схеми запуску АЦП (15), яка може приводитись до дії трьома різними способами (див.п.1.8), вхідна напруга подається в пам'ять-накопичувач (8).
3. Потому діючий рівень напруги підсилюється за допомогою програмованого підсилювача (14) і поступає в схему вибірки та збереження (9), що фіксує його рівень на вході АЦП (7) впродовж його перетворення в цифровий формат.
4. Про закінчення перетворення повідомляє сигнал ЕОС (End of convert) на виході АЦП.
5. Сигнал ЕОС використовується схемами керування передачею даних (4 – схема передавання даних за перериванням, 5 – схема передавання даних програмним способом).

Таблиця 2.1 – Основні блоки плати PCL-711

Блок	Пояснення
1	Дешифратор адреси
2	Регістр статусу
3	Буфер даних
4	Схема обміну за перериванням
5	Схема керування/статуса
6	Мультиплексор даних
7	АЦП
8	Пам'ять
9	Схема вибірка та збереження
10	Програмований таймер/лічильник
11	Дільник частоти
12	Генератор тактових імпульсів
13	Схема сканування каналів АЦП
14	Програмований підсилювач
15	Схема запуску
16	ЦАП
17	16 дискретних входів
18	16 дискретних виходів
19	Аналоговий мультиплексор

2.2 Навчальна програма PCL711.CPP

Навчальна програма PCL711.CPP (див. додаток А) була створена за допомогою СРП Borland C++ 5.2 і синтаксично сумісна із стандартом мови C++ **ISO/IEC/ANSI 14882:1998**.

Для того, щоб програма нормально трансьлювалася, необхідно щоб у каталозі проекту знаходилися файли:

EGAVGA.BGI – файл графічного драйверу BGI (**Borland Graphic Interface**) для відеоадаптерів EGA/VGA.

GRAPHICS.H – файл заголовку бібліотеки BGI.

GRAPHICS.LIB – файл бібліотеки BGI.

LITT.CHR – файл шрифту Little бібліотеки BGI.

Використання бібліотеки BGI звужує перелік компіляторів, що можуть бути використані для створення проекту. Але, як правило СРП різних виробників звичайно мають аналогічні бібліотеки графічних функцій.

2.3 Загальна організація PCL711.CPP

У навчальній програмі зсув адрес портів введення-виведення плати PCL-711 заданий за допомогою перелічування REG.

```
enum REG // Регистры платы PCL711
{
    CNT0 = 0, // Счетчик 0 (запись, чтение)
    CNT1 = 1, // Счетчик 1 (запись, чтение)
    CNT2 = 2, // Счетчик 2 (запись, чтение)
    CCOUNT = 3, // Упр. счетчиком (запись)
    DATAAD0 = 4, // Данные, мл.байт (запись-ЦАП, чтение-АЦП)
    DATAAD1 = 5, // Данные, ст.байт (запись-ЦАП, чтение-АЦП)
    DATADI0 = 6, // Дискр. вход, мл.байт (чтение)
    DATADI1 = 7, // Дискр. вход, ст.байт (чтение)
    RESET = 8, // Сброс прерывания (запись)
    SETGAIN = 9, // Коэффициент усиления АЦП (запись)
    CHANNEL = 10, // Выбор канала АЦП (запись)
    MODEINT = 11, // Режим и прерывание (запись)
    STARTAD = 12, // Программный запуск АЦП (запись)
    DATAD00 = 13, // Дискр. выход, мл.байт (запись)
    DATAD01 = 14, // Дискр. выход, ст.байт (запись)
};
```

Використані позначення регістрів надалі будемо використовувати для компактного представлення їх функціонального призначення (див. табл.1.3).

Для організації програмного обміну даними необхідно обов'язково задавати величину **тайм-ауту** – максимальної кількості невдалих спроб отримати дані від зовнішнього пристрою. В програмі PCL-711.CPP для цього використовується константа OVER.

```
const WORD OVER = 10000; // Тайм-аут
```

Ключовим об'єктом програми є клас PCL:

```
class PCL // Плата PCL711
{
```

```

private:
WORD base;           // Базовый адрес
WORD mode;           // Способ запуска АЦП и режим передачи
WORD gain;           // Усиление в каналах АЦП
WORD div;            //
WORD freq;           // Частота выборки
WORD reg [16];       // Регистры платы
WORD top[8];         // верх начала графика
WORD bot[8];         // низ начала графика
WORD weight;         // масштаб
int pos;             //
void showset (void); // отображение значений
void parset (void);  //
WORD keypressed (void); // вернуть скан-код нажатой клавиши

public:
WORD** data;         // кадр данных
WORD show;           // Способ отображения на экране
WORD channels;       // Всего каналов АЦП
void defboard (void); // настройки платы по умолчанию
WORD tuneboard (void); // ручная настройка платы
void resume (void);  // отображение регистров
void init (void);    // Подготовка подсистемы отображения
void protocol (WORD); // выполнить протокол
void showdata (void); // отображение кадра
};

```

На початку роботи програма виводить довідкову інформацію і надає користувачу можливість вибрати режим роботи:

-----			-----		
#	reg RW	val	#	reg RW	val

300	CNT0	++ 00	301	CNT1	++ 00*
302	CNT2	++ 00*	303	CCOUNT	-+ 00
304	DATAAD0	++ 00	305	DATAAD1	++ 00
306	DATADI0	+- 00	307	DATADI1	+- 00
308	RESET	-+ 00	309	SETGAIN	-+ 00
30a	CHANNEL	-+ 00	30b	MODEINT	-+ 00
30c	STARTAD	-+ 00	30d	DATAD00	-+ 00
30e	DATAD01	-+ 00			

```

Base address ..... 300      AD start ..... program
Data transfer .... program  AD channels ..... 1
AD gain ..... x1           Frequency, Hz .... 500

```

Show mode protocol

HANDLERS OF SETUP:

1 - Base address; 2 - Board mode; 3 - AD gain; 4 - AD channels;

5 - Show mode; 6 - Frequency; 9 - Exit;

Режим роботи настраюється за допомогою цифрових клавіш (табл.2.2).

Таблиця 2.2

Клавіша	Настройка
1	Вибір базової адреси
2	Спосіб запуску АЦП (програмний, зовнішнім сигналом або за допомогою таймера)
3	Коефіцієнт підсилення сигналу АЦП
4	Кількість каналів АЦП, що сканується
5	Режим відображення (протокол або графік 640x480x16)
6	Частота перетворення АЦП
9	Вихід
7, 8, 0	Продовження

Для вивчення логічного протоколу, що використовується під час будь-якого способу вмикання АЦП, зручно користуватися режимом відображення "протокол". Надалі результати роботи програми в цьому режимі наводяться для пояснення алгоритму взаємодії ПК з платою PCL-711.

2.4 Організація програмного запуску АЦП

Програмний запуск АЦП (див.п.1.2) виконується за допомогою процедури, що наведена в листингу 2.1.

Листинг 2.1

```

////////////////////////////////////
// Програмний запуск АЦП
////////////////////////////////////
void PRG_start (void)
{
WORD i, k;
WORD high, low, value;
for (k=0,i=0; i<channels; i++)
{
    outportb(base+CHANNEL, i);
    outportb(base+STARTAD, 0);
    delay(div);          // ЗАТРИМКА !!!
    do
    {
        k++;
        high = inportb(base+DATAAD1);
    }
    while ( (high & 0x10)!=0 && k<OVER);
    high = inportb(base+DATAAD1) & 0xf;
    low = inportb (base+DATAAD0) & 0xff;
    value = (high<<8)+low;
}
}

```

```

    }
}

```

Наприклад, якщо сканується чотири канали АЦП, то логічний протокол виглядає таким чином:

Output 0h to SETGAIN (BASE+9).	Set AD gain.
Output 00h to MODEINT (BASE+11).	Program start AD. Program data transfer.
Output 0 to CHANNEL (BASE+10).	Change AD0 channel.
Output 0 to STARTAD (BASE+12).	Start AD.
Check DRDY bit. Total attempts 1.	
Input 08h from DATAAD1 (BASE+5).	Input high 4 bits of AD result.
Input 12h from DATAAD0 (BASE+4).	Input low 8 bits of AD result.
Channel AD0 result is 0812h.	
.....	
Output 3 to CHANNEL (BASE+10).	Change AD3 channel.
Output 0 to STARTAD (BASE+12).	Start AD.
Check DRDY bit. Total attempts 4.	
Input 04h from DATAAD1 (BASE+5).	Input high 4 bits of AD result.
Input 63h from DATAAD0 (BASE+4).	Input low 8 bits of AD result.
Channel AD3 result is 0463h.	

Алгоритм 2.1 – Програмний запуск АЦП

1. В регістр SETGAIN (BASE+9) записується величина підсилення аналогового сигналу. Рівень входньої напруги, що надходить від плати ГПП не перевищує 3 В. Тому в SETGAIN записується 0 (коефіцієнт підсилення дорівнює x1).
2. В регістр MODEINT (BASE+11) записується цифра 0 – режим програмного способу обміну та програмний старт АЦП.
3. В регістр CHANNEL (BASE+10) записується номер каналу аналогового введення, що сканується.
4. В регістр STARTAD (BASE+12) записується будь-яке число. Запуск АЦП.
5. Виконується очікування зміни значення біту DRDY (біт D4 регістру DATAD1) з 1 на 0. Якщо виникне переповнення лічильника тайм-ауту – аварійний вихід (перехід до п.8).
6. Якщо біт DRDY дорівнює 0, то із регістра DATAD1 (BASE+5) спочатку отримують 4 старших розряди АЦП, а потім із регістра DATAD2 (BASE+4) отримують молодші 8 розрядів (розрядність АЦП 12 біт).
7. Якщо треба продовжувати, то перехід до п.3. Інакше – перехід до п.8.
8. Кінець.

На рис.2.2 наведені результати роботи в режимі відображення "graphics 640x480x16".

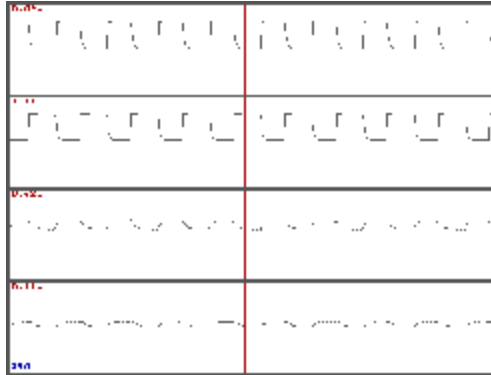


Рисунок 2.2 – Результати спостереження

Переміщення укажчика (червона вертикальна смужка на графіку) виконується за допомогою клавіш <LEFT>, <RIGHT> (грубе настроювання), або <CTRL>+<LEFT>, <CTRL>+< RIGHT > (точне настроювання).

2.5 Організація запуску АЦП зовнішнім сигналом

Запуск АЦП зовнішнім сигналом можливо виконати тільки тоді, коли з'єднувачі CN3 і CN4 поєднані за допомогою кабелю (забезпечена можливість подати на вхід D/I0 (контакт 1 з'єднувача CN4, див.п.1.2) сигнал старту АЦП (вихід D/O0, контакт 1 з'єднувача CN3)). Запуск АЦП зовнішнім сигналом виконується за допомогою процедури, що наведена в листингу 2.2.

Листинг 2.2

```

/////////////////////////////////////////////////////////////////
// Запуск АЦП зовнішнім сигналом
/////////////////////////////////////////////////////////////////
void EXT_start (void)
{
WORD i, k;
WORD high, low, value;
for (k=0,i=0; i<channels; i++)
{
    outputb(base+CHANNEL, i);
    outputb(base+DATAD00, 0);
    outputb(base+DATAD00, 1);
    delay(div);                // ЗАТРИМКА !!!
    do
    {
        k++;
        high = inportb(base+DATAAD1);
    }
}
}

```

```

    }
    while ( (high & 0x10)!=0 && k<OVER);
    high = inportb(base+DATAAD1) & 0xf;
    low = inportb (base+DATAAD0) & 0xff;
    value = (high<<8)+low;
  }
}

```

Наприклад, якщо сканується чотири канали АЦП, то логічний протокол виглядає таким чином:

```

Output 0h to SETGAIN (BASE+9).    Set AD gain.
Output 02h to MODEINT (BASE+11).  Extern start AD. Program data
transfer.
Output 0 to CHANNEL (BASE+10).    Change AD0 channel.
Output 0 to DATADO0 (BASE+13).    Initialize extern start AD.
Output 1 to DATADO0 (BASE+13).    Start AD by extern signal (CN4:1).
Check DRDY bit. Total attempts 1.
Input 08h from DATAAD1 (BASE+5). Input high 4 bits of AD result.
Input 2ah from DATAAD0 (BASE+4). Input low 8 bits of AD result.
Channel AD0 result is 082ah.
.....
Output 3 to CHANNEL (BASE+10).    Change AD3 channel.
Output 0 to DATADO0 (BASE+13).    Initialize extern start AD.
Output 1 to DATADO0 (BASE+13).    Start AD by extern signal (CN4:1).
Check DRDY bit. Total attempts 4.
Input 04h from DATAAD1 (BASE+5). Input high 4 bits of AD result.
Input 26h from DATAAD0 (BASE+4). Input low 8 bits of AD result.
Channel AD3 result is 0426h.

```

Алгоритм 2.2 – Запуск АЦП зовнішнім сигналом

1. В регістр SETGAIN (BASE+9) записується величина підсилення аналогового сигналу. Рівень входньої напруги, що надходить від плати ГПП не перевищує 3 В. Тому в SETGAIN записується 0 (коефіцієнт підсилення дорівнює x1).
2. В регістр MODEINT (BASE+11) записується цифра 02h – режим програмного способу обміну та старт АЦП зовнішнім сигналом.
3. В регістр CHANNEL (BASE+10) записується номер каналу аналогового введення, що сканується.
4. В регістр DATADO0 (BASE+13) записується 0. Підготовка до запуску АЦП.
5. В регістр DATADO0 (BASE+13) записується 1. Запуск АЦП переднім фронтом сигналу D/O0.
6. Виконується очікування зміни значення біту DRDY (біт D4 регістру DATAD1) з 1 на 0. Якщо виникне переповнення лічильника тайм-ауту – аварійний вихід (перехід до п.8).

7. Якщо біт DRDY дорівнює 0, то із регістра DATAD1 (BASE+5) спочатку отримують 4 старших розряди АЦП, а потім із регістра DATAD2 (BASE+4) отримують молодші 8 розрядів (розрядність АЦП 12 біт).
8. Якщо треба продовжувати, то перехід до п.3. Інакше – перехід до п.8.
9. Кінець.

На рис.2.3 наведені результати роботи в режимі відображення "graphics 640x480x16".

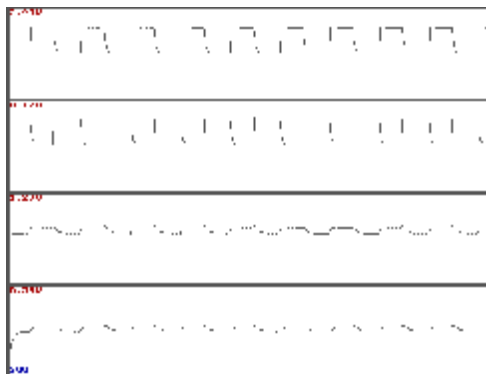


Рисунок 2.3 – Результати спостереження

Переміщення укажчика (червона вертикальна смужка на графіку) виконується за допомогою клавіш <LEFT>, <RIGHT> (грубе настроювання), або <CTRL>+<LEFT>, <CTRL>+< RIGHT > (точне настроювання).

2.6 Регістри програмованого таймера Intel 8253

На платі PCL-711 встановлена мікросхема 16-розрядного програмованого таймера/лічильника **Intel 8253** для завдання часу запуску. Вона забезпечує три незалежні канали таймера/лічильника (канали "Лічильник 0", "Лічильник 1" і "Лічильник 2" відповідно). Для розширення діапазону настройки тактової частоти є можливість сумісного використання Лічильника 1 і Лічильника 2 для завдання 32-розрядного дільника частоти (рис. 2.4).

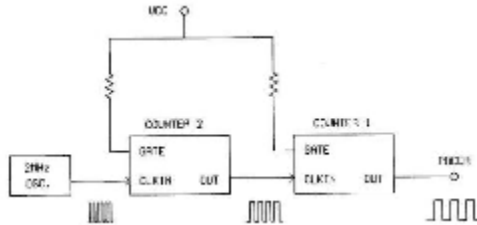


Рисунок 2.4 – Використовування дільника частоти таймера/лічильника Intel 8253

Регістри мікросхеми Intel 8253 доступні через чотири порти уведення-виведення (BASE+0, BASE+1, BASE+2, BASE+3):

BASE+0:	Лічильник 0	(Читання - Запис)
BASE+1:	Лічильник 1	(Читання - Запис)
BASE+2:	Лічильник 2	(Читання - Запис)
BASE+3:	Керування лічильником	(тільки Запис)

2.7 Регістр керування лічильником

BASE+3: Регістр керування лічильником (Запис)

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

Таблиця 2.3 – Вибір лічильника (біти SC0 і SC1)

SC1	SC0	Лічильник
0	0	0
0	1	1
0	0	2
1	1	заборонено

Таблиця 2.4 – Вибір операції читання - записи (біти RW0 і RW1)

RW1	RW0	Операція
0	0	Фіксація
0	0	Читання – запис LSB
0	1	Читання – запис MSB
1	1	Читання – запис LSB, потім читання – запис MSB

Таблиця 2.5 – Вибір операції читання - записи (біти RW0 і RW1)

M2	M1	M0	Режим
0	0	0	0 = Рахунок і зупинка
0	0	1	1 = Програмований однократний імпульс
X	1	0	2 = Генератор темпу
X	1	1	3 = Генератор прямокутних імпульсів
1	0	0	4 = Строб, що програмно генерується
1	0	1	5 = Апаратний строб, що генерується

Таблиця 2.6 – Режим рахунку (двійковий або двійково-десятковий)

BCD	Вид лічильника
0	16-розрядний двійковий лічильник (0 - 65535)
1	4-х декадний двійково-десятковий лічильник (0 – 9999)

За умовчанням використовується двійковий лічильник.

2.8 Режими роботи таймера/лічильника

Мікросхема Intel 8253 може працювати в одному з шести режимів (Режим 0 – Режим 5). Для настройки часу запуску Лічильник 1 і Лічильник 2 повинні бути запрограмовані для роботи в Режимі 2 (генератор частоти).

Рахунок і зупинка. На початку роботи в Режимі 0 на виході лічильника встановлюється низький логічний рівень. В рахунковий регістр заноситься величина часового періоду, і лічильник починає зворотний відлік. По закінченню періоду рахунку на виході лічильника встановлюється високий логічний рівень. Він залишатиметься високим до тих пір, поки вибраний лічильник не буде завантажений знову. Після завантаження зворотний рахунок продовжується і т.д. Якщо протягом роботи декрементного лічильника проводиться перезапис даних в рахунковий регістр, то:

1. Запис першого байта зупиняє лічильник.
2. Запис другого байта запускає новий рахунок.

Програмований однократний імпульс. В цьому режимі низький логічний рівень на виході встановлюється по передньому фронту сигналу, що поступає на вхід Gate під час роботи лічильника. Після закінчення рахунку встановлюється логічна 1. Якщо протягом часового інтервалу, коли на виході встановлений логічний 0, в рахунковий регістр буде завантажено нове значення, то цей факт не впливає на тривалість імпульсу, що генерується. Поточне значення рахункового регістра може бути прочитано у будь-який час. Низький логічний рівень на виході залишається протягом всього періоду рахунку.

Генератор частоти. В Режимі 2 на виході таймера/лічильника низький логічний рівень утримується протягом одного періоду вхідного тактового сигналу. Часовий інтервал між сусідніми імпульсами задається в рахунковому регістрі. Він рівний числу вхідних тактових сигналів. Завантаження рахункового регістра у момент роботи не приводить до зміни періоду проходження імпульсів.

Генератор прямокутних імпульсів. Цей режим подібний Режиму 2, за винятком того, що високий логічний рівень на виході тримається протягом першої половини заданого в рахунковому регістрі періоду. Протягом другої половини часового періоду встановлюється низький логічний рівень. Як тільки лічильник досягає останнього відліку, він завантажується, і процес повторюється.

В загальному випадку значення в рахунковому регістрі зменшується на два по задньому фронту кожного тактового імпульсу. Високий логічний рівень на виході утримується протягом кожного $(N + 1)/2$ тактового імпульсу; низький рівень – протягом кожного $(N - 1)/2$ імпульсу.

Строб, що програмно генерується. В Режимі 4 на виході таймера/лічильника встановлюється високий логічний рівень. Після завантаження значення в рахунковий регістр, лічильник починає зворотний відлік. В кінці роботи, протягом тривалості останнього рахункового імпульсу, на виході тримається низький логічний рівень. Потім рівень логічної 1 відновлюється.

Якщо протягом роботи декрементного лічильника проводиться перезапис даних в рахунковий регістр, то нове значення буде завантажено на наступному тактовому імпульсі. Рахунок не починається, поки на вході Gate лічильника встановлений рівень логічного 0.

Строб, що апаратно генерується. В Режимі 5 лічильник запускається за переднім фронтом зовнішнього сигналу запуску. В кінці роботи, протягом тривалості останнього рахункового імпульсу, на виході тримається рівень логічної 1.

2.9 Програмування мікросхеми Intel 8253

Для визначення тактової частоти F таймера/лічильника використовується формула (2.1). Рисунок 2.4 пояснює її.

$$F = \frac{2\text{МГц}}{(C1 \cdot C2)}; \quad (2.1)$$

Програмування необхідної тактової частоти виконується в наступній послідовності.

Алгоритм 2.3 – Програмування мікросхеми Intel 8253

1. Записати код 074H в порт CCOUNT (BASE+3). Налаштування Лічильника 1 для роботи в Режимі 2.
2. Записати відповідні дані (16-розрядне двійкове число в діапазоні від 2 до 65535) в порт COUNT1 (BASE+1). Установка константи дільника C1 для Лічильника 1. Спочатку в BASE+1 заноситься молодший байт C1, а потім - старший байт C1.
3. Записати код 0B4H в порт CCOUNT (BASE+3). Налаштування Лічильника 2 для роботи в Режимі 2.
4. Записати відповідні дані (16-розрядне двійкове число в діапазоні від 2 до 65535) в порт COUNT2 (BASE+2). Установка константи дільника C2 для Лічильника 1. Спочатку в BASE+2 заноситься молодший байт C2, а потім - старший байт C2.

2.10 Організація запуску АЦП сигналом таймера

Запуск АЦП сигналом таймера виконується за допомогою процедури, що наведена в листингу 2.3.

Листинг 2.3

```

////////////////////////////////////
// Запуск АЦП сигналом таймера
////////////////////////////////////
void CNT_start (void)
{
    WORD i, k;
    WORD high, low, value;
    low = div & 0xff;
    high = div>>8;
    outportb(base+MODEINT, 0x04);
    outportb(base+CCOUNT, 0x74);
    outportb(base+CNT1, low);
    outportb(base+CNT1, high);
    outportb(base+CCOUNT, 0xB4);
    outportb(base+CNT2, low);
    outportb(base+CNT2, high);
    for (k=0,i=0; i<channels; i++)
    {
        outportb(base+CHANNEL, i);
        delay(1000/freq); // ЗАТРИМКА !!!
        do
        {
            k++;
            high = inportb(base+DATAAD1);
        }
    }
}

```

```

    while ( (high & 0x10)!=0 && k<OVER);
    high = inportb(base+DATAAD1) & 0xf;
    low = inportb (base+DATAAD0) & 0xff;
    value = (high<<8)+low;
}
}

```

Наприклад, якщо сканується чотири канали АЦП, то логічний протокол виглядає таким чином:

Output 0h to SETGAIN (BASE+9).	Set AD gain.
Output 04h to MODEINT (BASE+11).	Timer start AD. Program data transfer.
Output 74h to CCOUNT (BASE+3).	Change mode 2 for Counter 1.
Output 3fh to CNT1 (BASE+1).	Frequency divider of counter 1.
Output B4h to CCOUNT (BASE+3).	Change mode 2 for Counter 2.
Output 3fh to CNT2 (BASE+2).	Frequency divider of counter 2.
Output 0 to CHANNEL (BASE+10).	Change AD0 channel.
Check DRDY bit. Total attempts 1.	
Input 00h from DATAAD1 (BASE+5).	Input high 4 bits of AD result.
Input 0h from DATAAD0 (BASE+4).	Input low 8 bits of AD result.
Channel AD0 result is 00h.	
Output 1 to CHANNEL (BASE+10).	Change AD1 channel.
Check DRDY bit. Total attempts 2.	
Input 08h from DATAAD1 (BASE+5).	Input high 4 bits of AD result.
Input 3ch from DATAAD0 (BASE+4).	Input low 8 bits of AD result.
Channel AD1 result is 083ch.	
.....	
Output 3 to CHANNEL (BASE+10).	Change AD3 channel.
Check DRDY bit. Total attempts 2228.	
Input 02h from DATAAD1 (BASE+5).	Input high 4 bits of AD result.
Input 61h from DATAAD0 (BASE+4).	Input low 8 bits of AD result.
Channel AD3 result is 0261h.	

Алгоритм 2.4 – Запуск АЦП сигналом таймера

1. В регістр SETGAIN (BASE+9) записується величина підсилення аналогового сигналу. Рівень входньої напруги, що надходить від плати ГПІ не перевищує 3 В. Тому в SETGAIN записується 0 (коефіцієнт підсилення дорівнює x1).
2. В регістр MODEINT (BASE+11) записується цифра 04h – режим програмного способу обміну та старт АЦП сигналом таймера.
3. В регістр CCOUNT (BASE+3) записується число 74H.
4. В регістр COUNT1 (BASE+1) записується дільник C1.
5. В регістр CCOUNT (BASE+3) записується число B4H.
6. В регістр COUNT2 (BASE+2) записується дільник C2.
7. В регістр CHANNEL (BASE+10) записується номер каналу аналогового введення, що сканується.

8. Виконується очікування зміни значення біту DRDY (біт D4 регістру DATAD1) з 1 на 0. Якщо виникне переповнення лічильника тайм-ауту – аварійний вихід (перехід до п.8).
9. Якщо біт DRDY дорівнює 0, то із регістра DATAD1 (BASE+5) спочатку отримують 4 старших розряди АЦП, а потім із регістра DATAD2 (BASE+4) отримують молодші 8 розрядів (розрядність АЦП 12 біт).
10. Якщо треба продовжувати, то перехід до п.3. Інакше – перехід до п.8.
11. Кінець.

На рис.2.5 наведені результати роботи в режимі відображення "graphics 640x480x16".

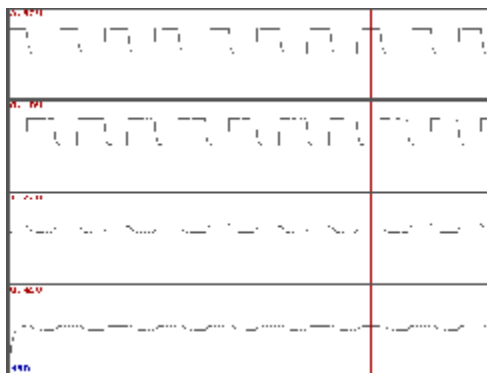


Рисунок 2.5 – Результати спостереження

Переміщення укажчика (червона вертикальна смужка на графіку) виконується за допомогою клавіш <LEFT>, <RIGHT> (грубе настроювання), або <CTRL>+<LEFT>, <CTRL>+< RIGHT > (точне настроювання).

2.11 Програмування системного таймеру ПК

Сумісний з IBM PC ПК містить програмований таймер, регістрова модель якого подібна розглянутій мікросхемі Intel 8253. Використання програмованого таймеру ПК має свої особливості.

Програмований таймер ПК підключений до лінії запиту переривання IRQ0 і виробляє переривання INT 8h з частотою 18,2 Гц. Під час ініціалізації BIOS встановлює свій обробник переривання таймера. Цей обробник кожного разу збільшує на одиницю поточне значення 4-байтової змінної, розташованої в області даних BIOS за адресою 0000:046Ch. Цю область звуть **лічильник таймера** (одиниця вимірювання часу – так звані "тіки"). Цей лічильник пере-

повнюється кожні 24 години з моменту запуску таймера. Після переповнення в лічильник таймера заноситься одиниця і рахунок відновлюється.

Канали програмованого таймеру ПК частково задіяні для виконання функцій, що забезпечують нормальну роботу комп'ютера.

Лічильник 0 використовується системним годинником ПК. Цей канал працює в режимі 3 і використовується як генератор імпульсів з частотою приблизно $1193180 \text{ Гц} / 65536 = 18,2 \text{ Гц}$. Ці імпульси викликають апаратне переривання INT 8h.

Лічильник 1 використовується для регенерації вмісту динамічної пам'яті комп'ютера. Вихід каналу OUT використовується для запиту до каналу прямого доступу DMA, який і виконує оновлення вмісту пам'яті.

Лічильник 2 підключений до гучномовця комп'ютера і може бути використаний для генерації різних звуків або музики. Канал використовує режим 3 таймеру.

Програмованому таймеру ПК відповідають чотири порти введення-виведення з наступними адресами:

- 40h - Лічильник 0;
- 41h - Лічильник 1;
- 42h - Лічильник 2;
- 43h - керуючий регістр.

Частота тактового генератора ПК - 1193180 Гц. Для того, щоб підвищити частоту викликів переривання INT 8h за допомогою Лічильника 0 таймера ПК (стандартно воно викликається 18,2 раз на секунду), необхідно зменшити дільник C0 Лічильника 0 (звичайно C0 = 65536).

Після закінчення вимірювань режим роботи Лічильника 0 необхідно обов'язково відновлювати.

Розглянемо випадок, коли частота відновлювання лічильника таймера BIOS повинна дорівнювати 25 кГц – максимальній частоті роботи плати.

1. Знайдемо величину дільника C0: $1193180/25000 \approx 48$. У шістнадцятирічній системі відліку $48 = 30\text{H}$. Старший байт (MSB) дільника C0 Лічильника 0 повинен дорівнювати нулю, а молодший байт (LSB) – 30H.
2. Звернувшись до довідникових матеріалів (п.2.6 - 2.8), сформуємо керуюче слово, що задає режим роботи Лічильника 0:

- біти D7:D6 = "00" – Лічильник 0;
- біти D5:D4 = "11" – запис/читання LSB, потім запис/читання MSB;
- біти D3:D1 = "011" – режим роботи – генератор тактових імпульсів;
- біт D0 = '0' – 16-розрядний двійковий лічильник.

3. Керуюче слово дорівнює $00110110 = 36\text{H}$.

Для програмування Лічильника 0 таймера необхідно виконати наступну послідовність дій.

1. Вивести в порт 43h слово 36H;
2. Необхідне значення дільника C0 посилається в порт 40h, причому спочатку виводиться молодший, а потім старший байти значення дільника C0.

Лічильник 0 таймера відразу почне виконувати необхідну функцію – генерувати переривання INT 8h з необхідною частотою.

Листинги 2.1-2.3 містять виклики бібліотечної функції delay(), за допомогою якою формується приблизна програмна затримка. Параметром функції delay() є число, що вказує величину затримки у мілісекундах ($1 \text{ мс} = 10^{-3} \text{ с}$). Цією функцією не можна створювати затримки, менші за 1 мс (максимальна частота вибірки – 1 кГц).

Листинги 2.4-2.7 містять функції, що виконують дії, за допомогою яких можна створювати необхідну програмну затримку, отримувати поточне значення лічильника таймера BIOS та змінювати частоту виклику переривання INT 8h.

Листинг 2.4

```

////////////////////////////////////
// Формування затримки
////////////////////////////////////
void mydelay (long delay)
{
    long begin, time;
    begin = ticks();
    do
        time = ticks();
        while (delay > begin - time);
}

```

Листинг 2.5

```

////////////////////////////////////
// Читання лічильника таймеру BIOS
////////////////////////////////////
long ticks (void)
{
    union REGS regs;
    long time;
    regs.x.ax = 0;
    int86 (0x1a, &regs, &regs);
    time = regs.x.cx;
    time = (time & 0000ffff) << 16;
    time += regs.x.dx;
    return time;
}

```

```
}
```

Листинг 2.6

```

////////////////////////////////////
// Настроювання Лічильника 0 системного таймера ПК
////////////////////////////////////
void settimer(long new)
{
    int cnt;
    outportb (0x43, 0x36);
    cnt = new & 0x00ff;
    outportb (0x40, cnt);
    cnt = (new & 0xff00) >> 8;
    outportb (0x40, cnt);
}

```

Листинг 2.7

```

////////////////////////////////////
// Читання дільника Лічильника 0 системного таймера
////////////////////////////////////
long gettimer(void)
{
    long div;
    int cnt;
    outportb (0x43, 0xb6);
    cnt = inportb (0x40) & 0x00ff;
    div = (long)cnt;
    div = (div&0x00ff) << 8;
    cnt = inportb (0x40) & 0x00ff;
    div += (long)cnt;
    return div;
}

```

Функція ticks() (див. листинг 2.5) для читання значення лічильника таймера BIOS використовує функцію 0 переривання INT 1Ah. Ця функція повертає у парі CX:DX чотири байти лічильника таймера BIOS.

2.12 Достовірність результатів спостереження

У загальному випадку під час організації програмного обміну з пристроєм зв'язку з об'єктом користувач повинен вміти.

1. Вимірювати фактичний час створення кадру даних.
2. Формувати програмну затримку.

В режимі відображення "graphics 640x480x16" програмою PCL711.CPP на екрані виводиться графік, що показує характер зміни вхідної напруги, що надходить з каналів аналогового введення, що скануються. Графік кожного

сигналу містить 600 точок. Для формування кадру даних потрібен час T , який дорівнює.

$$T = \frac{N \cdot 600}{F}; \quad (2.2)$$

де

N – кількість каналів, що скануються, F – частота вибірки вхідної напруги.

Звичайно фактичний час спостереження сигналу $T_1 \neq T$. Причиною цього ефекту є незалежні фактори (захоплення локальної шини зовнішнім пристроєм або контролером прямого доступу у пам'ять, нестабільність напруги електроживлення та ін.). Технічні характеристики звичайного ПК суттєво відрізняються від параметрів промислових інформаційно-керуючих обчислювальних систем.

Абсолютна похибка $\Delta T = |T_1 - T|$ фактично є мірою за допомогою якої можна оцінити достовірність результатів спостереження.

Листинги 2.8, 2.9 містять функції, що дозволяють оцінювати достовірність даних спостереження, якщо запуск АЦП плати PCL-711 виконується із програми або за допомогою зовнішнього сигналу.

Листинг 2.8

```

////////////////////////////////////
// функція обчислення абсолютної похибки
////////////////////////////////////
WORD getdelta (int mode, int time, int N)
{
    long delay, old, T1, T;
    int begin;
    double delta;

    old = gettimer();          // збереження старого дільника таймеру ПК
    settimer(48);              // 48 - максимальний дільник таймеру ПК
    T = (time*48)/1193180L;    // час спостереження в тіках (див.п.2.11)

    outportb(base+SETGAIN, x1); // коефіцієнт підсилення АЦП
    outportb(base+MODEINT, mode); // налаштування способу старту АЦП
    begin = ticks();           // початкове значення лічильника таймеру
    getframe (mode, N, 0);      // отримання кадру даних
    T1 = ticks() - begin;       // фактичний час спостереження
    delay = (T1-T)/(600*N);     // обчислення програмної затримки

    if (delay >= 0)
    {
        begin = ticks(); // початкове значення лічильника таймеру
        getframe (mode, N, delay); // отримання кадру даних
        T1 = ticks() - begin; // фактичний час спостереження
    }
}

```

```

// обчислення похибки часу спостереження
delta = abs(double((T1-T)*1193180L)/double(48));

cout << "Acquisition time T = " << T << "\n";
cout << "Acquisition time T1 = " << T1 << "\n";
cout << "Delay = " << delay << "\n";
cout << "Delta = " << delta << " sec\n";
}
else
    cout << "Error: delay is not positive.\n";

settimer(old);          // відновлення дільника таймеру ПК
}

```

Листинг 2.9

```

////////////////////////////////////
// Функція отримання кадру даних
////////////////////////////////////
WORD getframe (int mode, int N, long delay)
{
    WORD i, k;
    WORD high, low, value;
    for (j=0; j<600; j++)
    for (k=0,i=0; i<channels; i++)
    {
        outportb(base+CHANNEL, i); // вибір каналу АЦП
        switch (mode)
        {
            case PRG_PRG:          // програмний запуск АЦП
                outportb(base+STARTAD, 0);
                break;
            case EXT_PRG:          // запуск АЦП зовнішнім сигналом
                outportb(base+DATAD00, 0);
                outportb(base+DATAD00, 1);
                break;
        }
        mydelay(delay);
    }
    do
    {
        k++;
        high = inportb(base+DATAAD1);
    }
    // читання даних АЦП
    while ( (high & 0x10)!=0 && k<OVER);
    if (k>OVER)
        cout << "Error: time-out !!\n";
    else
    {
        high = inportb(base+DATAAD1) & 0xf;
        low = inportb (base+DATAAD0) & 0xff;
        value = (high<8)+low;
        data[i][j] = value;
    }
}

```

```

}
}
}

```

2.13 Завдання на роботу

1. Вивчити структурно-функціональну організацію плати PCL-711 і послідовність отримання даних від каналів аналогового уведення.
2. Вивчити логічний протокол програмного обміну за трьох різних способах запуску АЦП.
3. Вивчити правила використання таймеру ПК для створювання програмної затримки.
4. Вивчити методику оцінювання достовірності даних спостереження.
5. За останньою цифрою номера студентського квитка визначте вихідні данні на практичну частину лабораторної роботи (див. табл.2.7).
6. Розрахуйте T , C_0 .
7. Створіть файл PCL711N.CPP, куди скопіюйте вміст Додатку А (N – це номер індивідуального завдання).
8. Внесіть зміни у програму PCL711N.CPP. Для цього потрібно додати до переліку методів класу PCL функції `getdelta()`, `getframe()`, `gettimer()`, `settimer()`, `ticks()`, `mydelay()` і змінити функцію `main()` програми таким чином, щоб можна було отримати величину абсолютної похибки ΔT . Функцію `getframe()` можна скоротити – індивідуальне завдання передбачає використання програмою тільки одного способу запуску АЦП (програмний або зовнішнім сигналом).
9. Виконайте компіляцію програми. Наприклад, з командного рядка для цього можна використовувати команду `bcc PCL711N.cpp graphics.lib`.
10. Якщо є помилки в програмі – виправте їх.
11. Працюючу програму покажіть викладачеві.
12. Зробіть висновки.

Таблиця 2.7 – Індивідуальні завдання

Варіант	Каналів АЦП	Запуск АЦП	Час спостереження, мс
0	2	програмний	600
1	4	зовнішній	1200
2	3	зовнішній	1000
3	3	програмний	600
4	2	зовнішній	800
5	4	програмний	1500
6	4	зовнішній	1200
7	3	програмний	1000
8	2	зовнішній	500

9	3	програмний	900
---	---	------------	-----

2.14 Зміст звіту

1. Титульний лист. Мета роботи.
2. Короткий конспект – опис регістрової моделі мікросхеми Intel 8253.
3. Опис організації програми PCL711N.CPP.
4. Розрахунок T, C0, ΔT.
5. Висновки.

2.15 Контрольні запитання

1. Призначення і функціонально-структурна організація плати PCL-711.
2. Технічні характеристики каналу аналогового уведення, типова схема підключення.
3. Регістрова модель плати PCL-711. Які регістри по різному використовуються в режимі читання і режимі запису даних?
4. Три способи запуску АЦП плати PCL-711.
5. Логічний протокол програмного обміну даними, якщо використовується програмний спосіб запуску АЦП.
6. Логічний протокол програмного обміну даними, якщо використовується спосіб запуску АЦП сигналом таймеру.
7. Логічний протокол програмного обміну даними, якщо використовується спосіб запуску АЦП зовнішнім сигналом.
8. Регістрова модель мікросхеми програмованого таймеру Intel 8253.
9. Системний таймер ПК. Які функції виконують лічильники системного таймеру?
10. Лічильник таймеру BIOS. Принцип функціонування служби системного часу ПК.

ЛІТЕРАТУРА

1. PCL-711B PC-MultiLab Card. User manual.: Advantech.- 1993.
2. PCL-711B PC-MultiLab Card. Programmer manual.: Advantech.- 1993.
3. Эрглис К.Э. Интерфейсы открытых систем.- М.: Горячая линия-Телеком, 2000.- 256 с.
4. Хвощ С.Т. и др. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник/ С.Т. Хвощ, Н.Н. Варлинский, Е.А. Попов; Под общ. ред. С.Т. Хвоща.- Л.: Машиностроение, 1987.- 640 с.
5. Касаткин А.И. Профессиональное программирование на языке Си. Управление ресурсами: Справ. пособие. - Мн.: Выш. шк., 1992. - 432 с.
6. Фролов А.В., Фролов Г.В. Аппаратное обеспечение IBM PC. - М.: ДИАЛОГ-МИФИ, 1992.- 208 с.
7. Зубков С.В. Assembler для DOS, Windows и Unix.- М.:ДМК, 1999.- 640 с.
8. Ю.В. Воскобойник, О.О. Голдобін. Методичні вказівки до лабораторних робіт з дисципліни „Інтерфейси та пристрої зв'язку з об'єктом”- ЗНТУ, 2004