Разработка микроконтроллерной программы «EPOS8-LAS» для макета сенсорного блока полиграфа «ЭПОС-8»

Отчет

Заказчик: ЗАО «Группа Эпос»

Исполнитель (автор): Александр Бабиков

Москва 2008

Введение

По заказу ЗАО «Группа Эпос» (далее – Заказчика) в период с 16 по 24 мая 2008 г Исполнителем была осуществлена разработка микроконтроллерной программы «EPOS8-LAS» для макета сенсорного блока полиграфа «ЭПОС8».

Результаты разработки, помимо собственно микроконтроллерной программы, включают в себя специально разработанную программу для ПК, предназначенную для демонстрации работоспособности созданной микропрограммы, а также документальное обеспечение.

Настоящий отчет является приложением к переданным Заказчику результатам разработки.

Далее по тексту разработанная микроконтроллерная программа именуется «МК-программа», разработанная программа для ПК именуется «ПК-программа».

В данном отчете приводятся:

1. Перечень результатов разработки, переданных Заказчику.
2. Описание протокола взаимодействия МК-программы и ПК-программы.
3. Блок-схема и описание алгоритма МК-программы.
4. Исходный текст МК-программы (на языке ASM51).
5. Объектный код МК-программы (в формате Extended Intel Hex).
6. Инструкция по работе с ПК-программой для макета сенсорного блока «Эпос-8».

Содержание

Задание

Задача разработки состояла в следующем:

Требуется разработать МК-программу для МК предоставленного Заказчиком макета сенсорного блока полиграфа «ЭПОС-8». МК-программа управляет цифровыми преобразователями (ЦАП, АЦП) и мультиплексорами сенсорного блока, осуществляя опрос восьми датчиков (каналов) полиграфа. Оцифрованные результаты опроса без запроса от получателя (ПК) периодически передаются в ПК (в ПК-программу) по каналу UART (RS232). ПК может управлять параметрами усилителей сенсорного блока путем задания значений УСИЛЕНИЯ (GAIN) и СМЕЩЕНИЯ (BIAS) для каждого датчика (канала) выдачей соответствующих команд.

Протокол обмена информацией между МК и ПК, по требованию Заказчика, должен быть максимально приближен к протоколу, предоставленному Заказчиком (см.Приложение).

Результаты разработки, по требованию Заказчика, должны включать в себя: исходный текст МК-программы; описание протокола обмена информацией между МК- и ПК-программами; ПК-программу, позволяющую убедится в работоспособности созданной МК-программы.

Перечень результатов разработки

В соответствии с требованиями Заказчика, были разработаны: МК-программа «EPOS8-LAS», документальное обеспечение, тестовая ПК-программа.

Результаты разработки были переданы Заказчику после демонстрации их работоспособности и соответствия заданию.

Переданные результаты разработки включают в себя:

1. Файл с исходным текстом МК-программы на языке ASM51 (`epos8las.a51`).
2. Файл с объектным кодом МК-программы в формате Extended Intel Hex (`epos8las.hex`).
3. Один микроконтроллер AT89C4051 с прошитым в PROM файлом `epos8las.hex`.
4. Файл с ПК-программой (для WinXP/Vista), предназначенной для демонстрации работоспособности макета сенсорного блока под управлением разработанной МК-программы (`epos8las.exe`).
5. Файл с описанием протокола обмена информацией МК «EPOS8-LAS» и ПК (`epos8las.doc`).

Описание протокола обмена информацией между МК и ПК

Введение

Здесь приводится описание протокола обмена информацией между микроконтроллером (МК) под управлением разработанной МК-программы «EPOS8-LAS» и персональным компьютером (ПК). Настоящее описание создано в рамках разработки МК-программы «EPOS8-LAS» и является одним из результатов этой разработки.

Заказчиком было предоставлено описание протокола (см. Приложение), реализованного ранее в ПК-программе «ЭПОС 8», и поставлена задача максимизировать обратную совместимость протокола разработанной МК-программы «EPOS8-LAS» с этим протоколом. Предоставленный Заказчиком протокол ниже будем именовать *протокол Заказчика*.

Формат данных, передаваемых из МК «EPOS8-LAS» в ПК, полностью описывается протоколом Заказчика. Передача команд из ПК в МК также описывается протоколом Заказчика, за исключением вновь (дополнительно) введенных команд СТАРТ и СТОП.

Рекомендации по адаптации предоставленной Заказчиком ПК-программы «ЭПОС 8» к работе по описанному здесь протоколу приведены в соответствующем разделе настоящего описания.

1. Информация, посылаемая из МК в ПК

МК «EPOS8-LAS», будучи включенным, может находиться в одном из двух состояний: либо не отсылать в ПК никаких данных, либо отсылать в ПК пачки данных, формат которых задается в протоколе Заказчика.

При включении МК инициализирует свои внутренние регистры; один из них – флаг разрешения опроса АЦП, который сбрасывается. Если флаг разрешения опроса АЦП установлен, МК выполняет цикл, состоящий в опросе АЦП и посыле пачки данных в ПК. Если же флаг сброшен, МК не опрашивает АЦП и не посылает ничего в ПК, а находится в режиме ожидания команды. Для установки/сброса флага разрешения опроса АЦП предназначены команды СТАРТ/СТОП (см.ниже).

Когда флаг разрешения опроса АЦП установлен, МК посылает в ПК пачки данных. Формат пачки данных процитируем по протоколу Заказчика:

«…блоки (пачки, пакеты) данных (26 байт) следующей структуры:

<table>
<tr><th>№ байта</th><th>Обозначение</th><th>Описание</th></tr>
<tr><td>1</td><td>N1</td><td rowspan="3">Номер пачки N = N1+N2*256+N3*256^2</td></tr>
<tr><td>2</td><td>N2</td></tr>
<tr><td>3</td><td>N3</td></tr>
<tr><td>4</td><td>H0</td><td rowspan="7">Значения АЦП по каналам.
H_i – старший байт: {bin: 0 # # # X X X X }, где ### - номер канала
Li – младший байт (младшие 8 бит данных)</td></tr>
<tr><td>5</td><td>L0</td></tr>
<tr><td>6</td><td>H1</td></tr>
<tr><td>7
…</td><td>L1
…</td></tr>
<tr><td>18</td><td>H7</td></tr>
<tr><td>19</td><td>L7</td></tr>
<tr><td>20</td><td>B</td><td>Значение последнего принятого МК из ПК байта</td></tr>
<tr><td>21</td><td>C</td><td>Байт состояния приемника МК:
<table><tr><td>0</td><td>0</td><td>N_1</td><td>N_0</td><td>0</td><td>0</td><td>I</td><td>F</td></tr></table>Бит F инвертируется, когда МК принимает очередной байт из ПК.
*Биты N1,N2,I пока неинформативны.</td></tr>
<tr><td>22 – 26</td><td></td><td>Строка-признак окончания пачки: «*EOP#» (звездочка, большие латинские буквы, решетка). Звездочка – 22 байт, «E» – 23 байт и т.д.
*Планируется отказаться от включения в пачку байтов №№ 22-26</td></tr>
</table>

».

2. Команды, выполняемые МК

Ниже приведено описание команд, выполняемых МК «EPOS8-LAS». Формат команд соответствует протоколу Заказчика: «Команды, принимаемые МК, трехбайтные, причем [если в МК флаг разрешения опроса АЦП установлен] после отправки в МК каждого байта необходимо дождаться пачку данных и проанализировать байты B и C (№№ 20 и 21). Первый байт содержит код команды и номер канала, второй и третий байт – аргумент(ы)». Передача команды начинается с первого байта.

Ниже формат каждой команды приводится в виде таблицы, первый столбец которой содержит первый байт команды, второй – второй, третий – третий. Знак «$» указывает на то, что следующее за ним число представлено в шестнадцатеричной системе счисления. Запись $XX означает любое число.

Перечень команд, выполняемых МК, с их описанием

1. СТАРТ

Команда СТАРТ устанавливает флаг разрешения опроса АЦП. Когда флаг разрешения опроса АЦП установлен, МК постоянно выполняет опросы АЦП и посыл пачек данных в ПК. Формат пачек данных не изменился по сравнению с протоколом Заказчика. По умолчанию (при включении питания) флаг разрешения опроса АЦП сброшен.

Формат команды:

$B0	$XX	$XX

Команда СТАРТ в протоколе Заказчика отсутствует.

2. СТОП

Команда СТОП сбрасывает флаг разрешения опроса АЦП. Когда флаг разрешения опроса АЦП сброшен, опрос АЦП и посыл пачек данных в ПК не выполняется; МК находится в режиме ожидания команды. По умолчанию (при включении питания) флаг разрешения опроса АЦП сброшен.

Формат команды:

$C0	$XX	$XX

Команда СТОП в протоколе Заказчика отсутствует.

3. БАЗА

Команда БАЗА устанавливает новое заданное значение базы (смещения) для заданного канала.

Формат команды:

$A0 + N	Б div $40	Б mod $40
N – номер канала [0..7]	Б – значение базы [0..4095], div – оператор целочисленного деления, mod – оператор взятия остатка от деления	

Команда БАЗА соответствует протоколу Заказчика.

4. УСИЛЕНИЕ

Команда УСИЛЕНИЕ устанавливает новое заданное значение усиления для заданного канала.

Формат команды:

$90 + N	У	У + $20
N – номер канала [0..7]	У – значение усиления [0..7]	

Команда УСИЛЕНИЕ соответствует протоколу Заказчика.

3. Рекомендации по адаптации ПК-программы «ЭПОС 8»

ПК-программа «ЭПОС 8», предоставленная Заказчиком, предназначена для работы с МК по протоколу Заказчика. Для того чтобы ПК-программа «ЭПОС 8» работала с МК «EPOS8-LAS» по описанному здесь протоколу, необходимо внести в нее дополнения.

При запуске ПК-программа «ЭПОС 8», очевидно, перебирает существующие в операционной системе и доступные COM-порты. Открыв очередной порт, «ЭПОС 8» читает из него данные, если таковые имеются. В том случае, если к данному порту подключен сенсорный блок с МК, работающим по протоколу Заказчика, «ЭПОС 8» считывает из порта несколько пачек данных, которые МК посылает в ПК без какого-либо запроса. Убедившись, что принятые данные по своей структуре соответствуют пачкам данных сенсорного блока, «ЭПОС 8» завершает поиск порта с сенсорным блоком.

МК «EPOS8-LAS» после включения не начинает без запроса посылать в ПК пачки данных. В этом случае нынешняя программа «ЭПОС 8» не опознает сенсорный блок. Цикл опроса портов программы «ЭПОС 8» следует несколько изменить. Если сейчас этот цикл работает по алгоритму «Открыть очередной порт и слушать», то можно – «Открыть очередной порт, отправить команду СТАРТ и слушать». Отметим, что подобное дополнение необходимо только в том случае, когда во время запуска «ЭПОС 8» в МК «EPOS8-LAS» флаг разрешения опроса АЦП сброшен.

Укрупненный алгоритм МК-программы

Описание алгоритма

МК-программа состоит из нескольких функциональных блоков:

1. Инициализация.
2. Основной цикл.
3. Подпрограммы управления цифровыми преобразователями и мультиплексорами.
4. Подпрограмма обработки полученных из ПК команд.

При включении МК выполняются операции блока «Инициализация», затем управление передается в «Основной цикл», из которого осуществляются вызовы подпрограмм управления цифровыми преобразователями и мультиплексорами и обработки команд из ПК.

Инициализация

Операции этого блока предназначены для настройки работы компонентов микроконтроллера, таких как таймеры-счетчики и приемо-передатчик UART, путем установки соответствующих битов регистров специального назначения МК. Также здесь выполняется установка начальных значений регистров состояния МК-программы, таких как счетчик пачек данных (опросов) и флаг разрешения опроса АЦП.

Разрешаются прерывания от UART, а сам приемо-передатчик настраивается на следующие параметры: 19200 8 1 N.

Счетчик пачек данных сбрасывается в 0. Флаг разрешения опроса АЦП сбрасывается.

Основной цикл

Внутри основного цикла выполняется ожидание команд из ПК и, если установлен флаг разрешения опроса АЦП, опрос датчиков и выдача данных в ПК.

В начале основного цикла проверяется, установлен ли флаг разрешения опроса АЦП. Если флаг установлен, выполняется опрос всех каналов и осуществляется передача пачки данных в ПК. Операции опроса каналов имеют следующую последовательность. Открывается цикл из восьми итераций (по числу датчиков). В каждой итерации из оперативной памяти МК считываются и устанавливаются значения смещения и усиления очередного канала и выполняется опрос АЦП, результаты которого

сохраняются в оперативной памяти. После восьми итераций в ПК отсылается инкрементированный номер пачки данных, извлеченные из оперативной памяти значения опросов АЦП, и строка-признак окончания пачки данных. Следует возврат на начало основного цикла.

Если флаг разрешения опроса не установлен, то опрос АЦП и передача пачки данных в ПК не выполняется.

Во время выполнения операций основного цикла из ПК могут прийти команды. Приходящие из ПК команды вызывают прерывание, которое обрабатывается соответствующей подпрограммой. Порядок посыла команд из ПК в МК задается протоколом, предоставленным Заказчиком.

Подпрограммы управления цифровыми преобразователями и мультиплексорами

Подпрограммы управления АЦП, ЦАП, коммутаторами каналов и усиления вызываются из основного цикла, если опрос датчиков разрешен установленным флагом разрешения опроса АЦП.

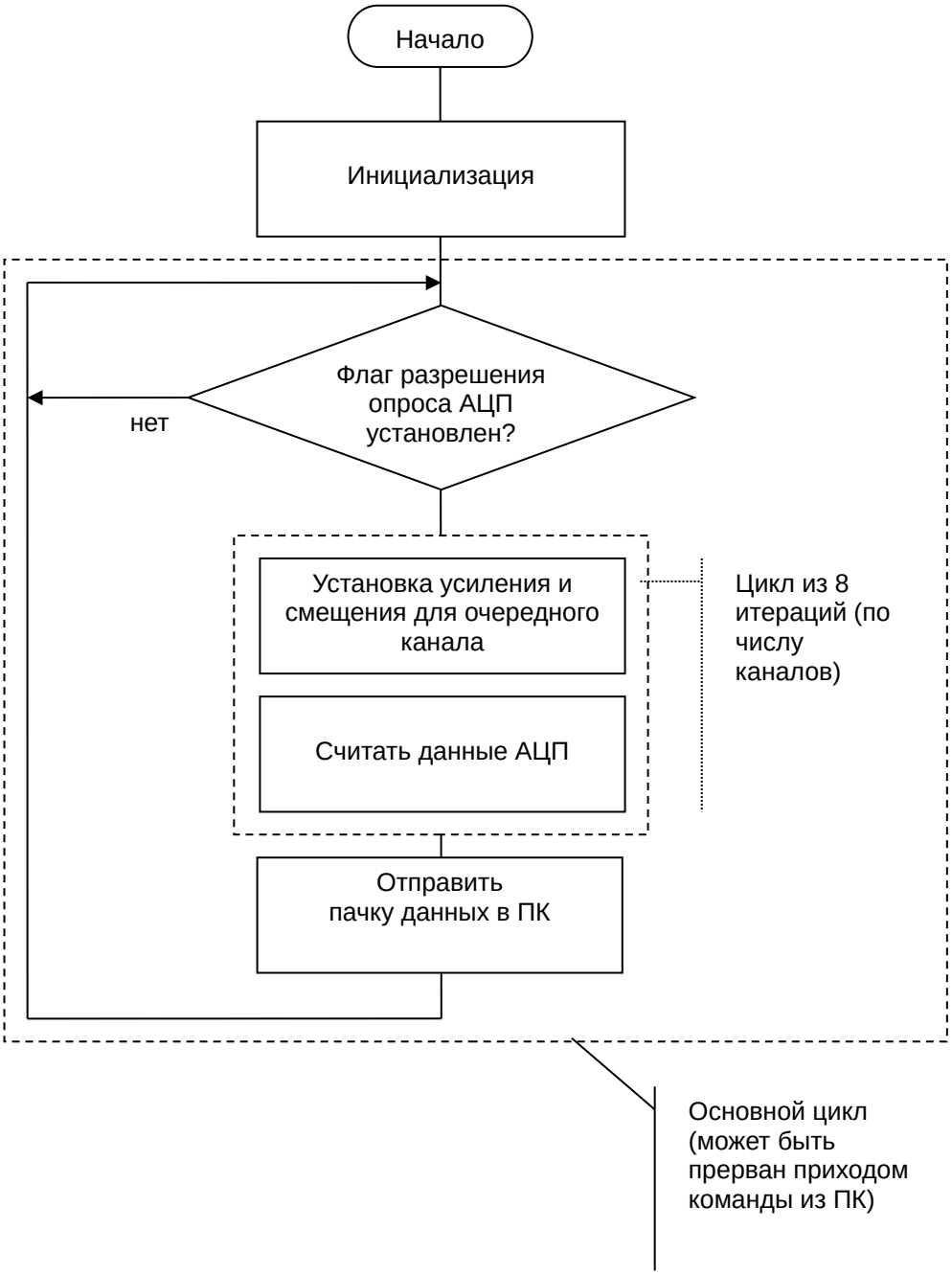
Алгоритмы управления цифровыми преобразователями и мультиплексорами определяются в спецификациях этих устройств и в отдельном описании не нуждаются.

Подпрограмма обработки полученных из ПК команд

Данная подпрограмма является подпрограммой обработки прерывания от UART и предназначена для приема и выполнения команд от ПК.

Приход каждого байта из ПК в МК сопровождается вызовом этой подпрограммы. После накопления трех байтов, пришедших последовательно друг за другом и составляющих команду, выполняется проверка кода команды (он содержится в первом байте команды) и, если код соответствует одному из допустимых (см. Перечень команд в Описании протокола), выполняется соответствующая команда. Если же код команды не определен в Протоколе, то пришедшая команда игнорируется.

Блок-схема алгоритма



Исходный текст МК-программы

```
; "EPOS8-LAS" FOR "EPOS-8" POLYGRAPH
; (C) ALEXANDER LUKYANOV, GPI, 2008.

      DC1_PINS    EQU   0B3H
      DC1_PIND    EQU   0B2H
      DC1_PINB    EQU   0B5H
      DC2_PINS    EQU   0B7H
      DC2_PINI    EQU   0B4H
      DC2_PINR    EQU   090H
      DC2_PINZ    EQU   091H
      DC2_PINB    EQU   0B5H
      DC3_PINS    EQU   094H
      DC3_PIND    EQU   093H
      DC3_CS            EQU   092H
      DC4_PINS    EQU   097H
      DC4_PIND    EQU   096H
      DC4_CS            EQU   095H
ORG   0
      AJMP  START
ORG   080H
      StringEndOfPack:
      DB    '*EOP#'
      STACKTOP          EQU   16
      BYTECOUNTER       EQU   32
      EXPECTEDIDENTIFIER      EQU   33
      LASTRECEIVEDCODE  EQU   34
      CHARBEGIN         EQU   '*'
      CHAREND                 EQU   '#'
      TIMERVALUE        EQU   040H
      ENABLE_POLLING          EQU   028H
      CORRECT_PASSWORD  EQU   029H
      CORRECT_SERIAL          EQU   02AH
      CORRECT_CHECKSUMM EQU   02BH
      EXPECTEDFLAG            EQU   02CH
      NOERRORFLAG       EQU   02DH
      TEMPBIT                 EQU   02EH
      APRM0_TABLETOP          EQU   38
      APRM1_TABLETOP          EQU   APRM0_TABLETOP+8
      BIASDATA_TABLETOP EQU   APRM1_TABLETOP+16
      DC1_H             EQU   BIASDATA_TABLETOP+16
      DC1_L             EQU   DC1_H+1
      PACKNUMBER0       EQU   DC1_L+1
      PACKNUMBER1       EQU   PACKNUMBER0+1
      PACKNUMBER2       EQU   PACKNUMBER0+2
      COMMANDCODE       EQU   PACKNUMBER2+1
      ARGUMENT2         EQU   COMMANDCODE+1
      ARGUMENT1         EQU   COMMANDCODE+2
ORG   100H
      TABLE_CHADDRESSES:
      DB    000B
      DB    001B
      DB    010B
      DB    011B
      DB    100B
      DB    101B
      DB    110B
      DB    111B
ORG   03H
      RETI
ORG   0BH
      RETI
ORG   013H
```

```
      RETI
ORG   01BH
      RETI
ORG   23H
      AJMP  SERIALINTERRUPT
ORG   200H
START:
      CLR   EA
      MOV   SP,   #STACKTOP
      MOV   IP,   #00010000B
      MOV   TCON, #00000000B
      MOV   TMOD, #00100001B
      MOV   87H,  #10000000B
      MOV   SCON, #01000000B
      MOV   TL1,  #0FDH
      MOV   TH1,  #0FDH
      SETB  TR0
      SETB  TR1
      SETB  TI
      SETB  REN
      MOV   IE,   #10010000B
      MOV   PACKNUMBER0,      #0
      MOV   PACKNUMBER1,      #0
      MOV   PACKNUMBER2,      #0
      SETB  NOERRORFLAG
      MOV   BYTECOUNTER,            #3
      MOV   EXPECTEDIDENTIFIER,     #032H
      MOV   LASTRECEIVEDCODE, #0
      ;SETB ENABLE_POLLING
      CLR   ENABLE_POLLING
      SETB  CORRECT_PASSWORD
      SETB  CORRECT_CHECKSUMM
      SETB  CORRECT_SERIAL
MAIN_LOOP:
      MOV   TH0,  #0
      MOV   TL0,  #0
      JB    ENABLE_POLLING,   POLLING0
JMP   LBLPAUSE
POLLING0:
      JNB   CORRECT_PASSWORD, SENDDATATOPC
      MOV   R4,   #255
      MOV   R3,   #8
      ONECHANAL:
      INC   R4
      MOV   A,    r4
      MOV   B,    #2
      MUL   AB
      ADD   A,    #APRM1_TABLETOP
      MOV   R0,   A
      MOV   A,    @R0
      INC   R0
      MOV   B,    @R0
      ACALL WRITE_DC1
      MOV   A,    R4
      MOV   DPTR, #TABLE_CHADDRESSES
      MOVC  A,    @A+DPTR
      ACALL       WRITE_DC3
      MOV   A,    R4
      ADD   A,    #APRM0_TABLETOP
      MOV   R0,   A
      MOV   A,    @R0
      ACALL LOAD_APRM0MP
      MOV   R0,   #128
      DC1:
      MOV   A,    #1
```

```
      DJNZ  ACC,  $
        DJNZ      R0,   DC1
      ACALL WRITE_DC2
      MOV   A,    R4
      MOV   B,    #2
      MUL   AB
      ADD   A,    #BIASDATA_TABLETOP
      MOV   R0,   A
      MOV   A,    DC1_H
      ANL   A,    #00FH
      SWAP  A
      ORL   A,    R4
      SWAP  A
      MOV   @R0,  A
      INC   R0
      MOV   @R0,  DC1_L
      DJNZ  R3,   ONECHANAL
SENDDATATOPC:
      CLR   REN
      ACALL INC_PACKNUMBER
SENDPACKNUMBER:
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, PACKNUMBER0
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, PACKNUMBER0+1
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, PACKNUMBER0+2
SENDADCTABLE:
      MOV   R0,   #BIASDATA_TABLETOP
      MOV   R3,   #8
GETONECHANNEL:
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, @R0
      INC   R0
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, @R0
      INC   R0
      DJNZ  R3,   GETONECHANNEL
SENDCURRENTCOMMANDSTATE:
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, LASTRECEIVEDCODE
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, EXPECTEDIDENTIFIER
SENDENDPACKSTRING:
      MOV   DPTR, #STRINGENDOFPACK
      ACALL SEND_STRING
      SETB  REN
LBLPAUSE:
      MOV   A,    TH0
      CLR   C
      SUBB  A,    TIMERVALUE
;JC   LBLPAUSE
      JMP   MAIN_LOOP
SEND_STRING:
      MOV   R3,   #0
SEND_STRING1:
      MOV   A,    R3
      MOVC  A,    @A+DPTR
```

```
      JNB   TI,   $
      CLR   TI
      MOV   SBUF, A
      INC   R3
      CJNE  A,    #CHAREND,   SEND_STRING1
      RET
WRITE_DC2:
      SETB  DC1_PIND
      CLR   DC1_PINS
        MOV B,    #2
cicle_2:
        CLR DC1_PINB
        SETB      DC1_PINB
        MOV C,    DC1_PIND
        RLC A
      DJNZ  B,    CICLE_2
        MOV A,    #0
        MOV B,    #4
cicle_high:
        CLR DC1_PINB
        SETB      DC1_PINB
        MOV C,    DC1_PIND
        RLC A
      DJNZ  B,    CICLE_HIGH
        MOV DC1_H,      A
      MOV   B,    #8
cicle_LOW:
        CLR DC1_PINB
        SETB      DC1_PINB
        MOV C,    DC1_PIND
        RLC A
      DJNZ  B,    CICLE_LOW
        MOV DC1_L,      A
        SETB      DC1_PINS
      RET
WRITE_DC1:
      PUSH  B
      SETB  DC2_PINS
      SWAP  A
      MOV   B,    #4
DC2_LOWbit:
        RLC A
        MOV DC2_PINI,   C
        CLR DC2_PINB
        SETB      DC2_PINB
      DJNZ  B,    DC2_LOWbit
      POP   ACC
      MOV   B,    #8
DC2_Lbit:
        RLC A
        MOV DC2_PINI,   C
      CLR   DC2_PINB
      SETB  DC2_PINB
      DJNZ  B,    DC2_Lbit
      CLR   DC2_PINS
      RET
WRITE_DC3:
      RRC   A
      MOV   DC4_PINS,   C
      RRC   A
      MOV   DC4_PIND,   C
      RRC   A
      MOV   DC4_CS,     c
      RET
LOAD_APRM0MP:
```

```
      RRC   A
      MOV   DC3_PINS,   C
      RRC   A
      MOV   DC3_PIND,   C
      RRC   A
      MOV   DC3_CS,     C
      RET
INC_PACKNUMBER:
      PUSH  ACC
      INC   PACKNUMBER0
      MOV   A,    PACKNUMBER0
      JNZ   RET_INCPACKNUMBER
      INC   PACKNUMBER0+1
      MOV   A,    PACKNUMBER0+1
      JNZ   RET_INCPACKNUMBER
      INC   PACKNUMBER0+2
RET_INCPACKNUMBER:
      POP   ACC
      RET
SERIALINTERRUPT:
      JB    TI,   INT_FROM_TI
      JB    RI,   INT_FROM_RI
      RETI
INT_FROM_TI:
      JB    RI,   INT_FROM_RI
      RETI
INT_FROM_RI:
      PUSH  ACC
      PUSH  B
      PUSH  000H
      PUSH    003H
      MOV   TEMPBIT,    C
      MOV   A,    SBUF
      CLR   RI
      MOV   LASTRECEIVEDCODE, A
      RLC   A
      JC    NEWCOMMAND
      MOV   A,    BYTECOUNTER
      CJNE  A,    #3,   ARGUMENTCAME
ERRORCAME:
      MOV   BYTECOUNTER,      #3
      CLR   NOERRORFLAG
      JMP   EXITFROMINT
ARGUMENTCAME:
      ADD   A,    #ARGUMENT2-1
      MOV   R0,   A
      MOV   @R0,  LASTRECEIVEDCODE
      MOV   A,    BYTECOUNTER
      DEC   A
      JZ    PERFORMCOMMAND
      MOV   BYTECOUNTER,      A
      JMP   EXITFROMINT
PERFORMCOMMAND:
      MOV   BYTECOUNTER,      #3
      SETB  NOERRORFLAG
      MOV   A,    COMMANDCODE
      ANL   A,    #007H
      MOV   R3,   A
      MOV   A,    COMMANDCODE
      ANL   A,    #0F0H
MAYAPRM1:
      CJNE  A,    #0A0H,      MAYAPRM0
      MOV   A,    R3
      MOV   B,    #2
      MUL   AB
```

```
      ADD   A,    #APRM1_TABLETOP
      MOV   R0,   A
      MOV   A,    ARGUMENT1
      RR    A
      RR    A
      MOV   @R0,  A
      INC   R0
      ANL   A,    #11000000B
      ORL   A,    ARGUMENT2
      MOV   @R0,  A
      JMP   EXITFROMINT
MAYAPRM0:
      CJNE  A,    #090H,      MAYSTART
      MOV   A,    R3
      ADD   A,    #APRM0_TABLETOP
      MOV   R0,   A
      MOV   @R0,  ARGUMENT1
      JMP   EXITFROMINT
MAYSTART:
      CJNE  A,    #0B0H,      MAYSTOP
      SETB  ENABLE_POLLING
      JMP   EXITFROMINT
MAYSTOP:
      CJNE  A,    #0C0H,      MAYELSE
      CLR   ENABLE_POLLING
      JMP   EXITFROMINT
MAYRESET:
      CJNE  A,    #0D0H,      MAYELSE
      AJMP  0
MAYELSE:
      JMP   EXITFROMINT
NEWCOMMAND:
      MOV   BYTECOUNTER,      #2
      SETB  NOERRORFLAG
      MOV   COMMANDCODE,      LASTRECEIVEDCODE
EXITFROMINT:
      CPL   EXPECTEDFLAG
      MOV   A,    BYTECOUNTER
      SWAP  A
      RRC   A
      RRC   A
      MOV   C,    NOERRORFLAG
      RLC   A
      MOV   C,    EXPECTEDFLAG
      RLC   A
      MOV   EXPECTEDIDENTIFIER,     A
      MOV   C,    TEMPBIT
      POP   003H
      POP   000H
      POP   B
      POP   ACC
      RETI
END.
```

Объектный код МК-программы

Объектный код МК-программs «EPOS8-LAS» приводится в формате Extended Intel Hex.

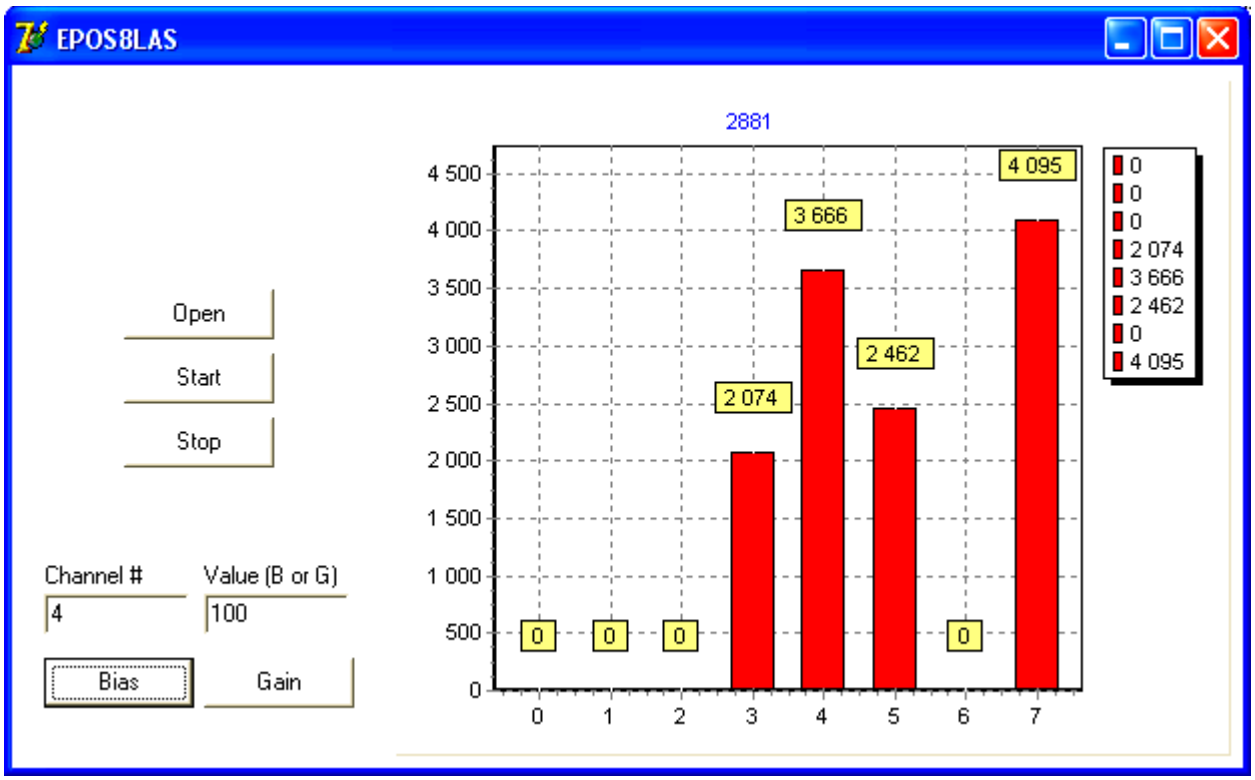
```
:020000004100BD
:050080002A454F50234A
:0801000000010203040506070DB
:0100030032CA
:01000B0032C2
:0100130032BA
:01001B0032B2
:0200230061700A
:10020000C2AF75811075B810758800758921758722
:1002100080759840758BFD758DFDD28CD28ED299EC
:10022000D29C75A8907550007551007552000D22D62
:1002300075200375213275220 0C228D229D22BD213
:100240002A758C00758A002028030202DA302941C1
:100250007CFF7B080CEC75F002A4242EF8E60886DF
:10026000F07125EC900100937149EC2426F8E671B9
:1002700053788074 01D5E0FDD8F951F1EC75F002A6
:10028000A4243EF8E54E540FC44CC4F608A64FDB38
:10029000C3C29C715D3099FDC2998550993099FD1A
:1002A000C2998551993099FDC299855299783E7BC2
:1002B000083099FDC2998699083099FDC2998699AE
:1002C00008DBEE3099FDC2998522993099FDC299DB
:1002D00085219990008051E1D29CE58CC3954041E5
:1002E000417B00EB933099FDC299F5990BB423F350
:1002F00022D2B2C2B375F002C2B5D2B5A2B233D522
:10030000F0F6740075F004C2B5D2B5A2B233D5F0E0
:10031000F6F54E75F008C2B5D2B5A2B233D5F0F6F7
:10032000F54FD2B322C0F0D2B7C475F0043392B403
:10033000C2B5D2B5D5F0F6D0E075F0083392B4C2AC
:10034000B5D2B5D5F0F6C2B7221392971392961391
:1003500092952213929413929313929222C0E005E5
:1003600050E55070080551E55170020552D0E02269
:10037000209904209805322098 0132C0E0C0F0C0D6
:1003800000C003922EE599C298F522334067E5201C
:10039000B40308752003C22D0203FD2453F8A622DE
:1003A000E520146005F5200203FD752003D22DE53C
:1003B000535407FBE55354F0B4A016EB75F002A4B8
:1003C000242EF8E5550303F60854C04554F60203FD
:1003D000FDB49009EB2426F8A6550203FDB4B00540
:1003E000D2280203FDB4C00AC2280203FDB4D00221
:1003F00001000203FD752002D22D852253B22CE5A7
:1004000020C41313A22D33A22C33F521A22ED00326
:07041000D000D0F0D0E03273
:00000001FF
```

Инструкция к тестовой ПК-программе

Тестовая ПК-программа `EPOS8LAS.EXE` предназначена для демонстрации работоспособности созданной МК-программы. ПК-программа работает с макетом сенсорного блока полиграфа «ЭПОС-8», предоставленного Заказчиком Исполнителю на время разработки МК-программы, и возвращенного Заказчику при передаче результатов разработки.

ПК-программа позволяет отсылать в МК все предусмотренные Протоколом команды и принимать выдаваемую МК информацию.

Окно ПК-программы приведено на рисунке:



Назначение кнопок:

1. Кнопка «Open» предназначена для выбора и открытия COM-порта, к которому подключен сенсорный блок.
2. Кнопка «Start» отсылает команду СТАРТ в сенсорный блок.
3. Кнопка «Stop» отсылает команду СТОП в сенсорный блок.
4. Кнопка «Bias» отсылает команду БАЗА в сенсорный блок. Номер канала (0--7) должен быть предварительно введен в поле «Channel #», значение БАЗЫ (0--4095) должно быть предварительно введено в поле «Value».

5. Кнопка «Gain» отсылает команду УСИЛЕНИЕ в сенсорный блок. Номер канала (0--7) должен быть предварительно введен в поле «Channel #», значение УСИЛЕНИЯ (0--7) должно быть предварительно введено в поле «Value».

На диаграмме в правой части окна программы выводятся значения, считываемые МК с АЦП и направляемые из МК в ПК. Каждый столбец диаграммы соответствует одному каналу (датчику). В заголовке диаграммы выводится значение счетчика пачек данных, также принимаемое из МК.

Для работы с ПК-программой необходимо выполнить следующие действия:

1. Подключить макет сенсорного блока полиграфа «ЭПОС-8» под управлением МК-программы «EPOS8-LAS» к ПК. (При необходимости установить драйверы виртуального COM-порта).
2. Запустить ПК-программу `EPOS8LAS.EXE` в среде Windows XP или Windows Vista.
3. Нажать кнопку «Open» в окне ПК-программы и выбрать COM-порт, к которому подключен сенсорный блок.
4. Нажать кнопку «Start», что означает отправку команды СТАРТ в МК. На диаграмме наблюдать результаты оцифровки показаний датчиков.
5. Для изменения СМЕЩЕНИЯ по какому-либо каналу ввести в поле ввода «Channel #» номер канала, а в поле ввода «Value» – значение СМЕЩЕНИЯ (0--4095). Нажать кнопку «Bias», что означает отправку команды БАЗА в МК.
6. Для изменения УСИЛЕНИЯ по какому-либо каналу ввести в поле ввода «Channel #» номер канала, а в поле ввода «Value» – значение УСИЛЕНИЯ (0--7). Нажать кнопку «Gain», что означает отправку команды УСИЛЕНИЕ в МК.
7. Для останова опроса нажатием кнопки «Stop» отправить в МК команду СТОП.

Приложение

«Протокол Заказчика»
(это описание предоставлено Заказчиком и явилось основой выполненной разработки)

Эпос. Сентябрь 2005.

Макет «Боец». Промежуточный отладочный вариант.

Протокол обмена информацией сенсорного блока (МК) и компьютера (ПК).

из МК в ПК

МК работает в активном режиме, т.е. выдача данных в ПК осуществляется без запроса со стороны ПК. После включения МК сразу начинает выдавать блоки (пачки, пакеты) данных (26 байт) следующей структуры:

№ байта	Обозначение	Описание
1	N1	Номер пачки N = N1+N2*256+N3*256^2
2	N2	
3	N3	
4	H0	Значения АЦП по каналам. H_i – старший байт: {bin: 0 # # # X X X X }, где ### - номер канала Li – младший байт (младшие 8 бит данных)
5	L0	
6	H1	
7	L1	
…	…	
18	H7	
19	L7	
20	B	Значение последнего принятого МК из ПК байта
21	C	Байт состояния приемника МК: 0 0 N_1 N_0 0 0 I F Бит F инвертируется, когда МК принимает очередной байт из ПК. *Биты N1,N2,I пока неинформативны.
22 – 26		Строка-признак окончания пачки: «*EOP#» (звездочка, большие латинские буквы, решетка). Звездочка – 22 байт, «E» – 23 байт и т.д. *Планируется отказаться от включения в пачку байтов №№ 22-26

В секунду МК выдает прибл. 35 пачек.

из ПК в МК

Команды, принимаемые МК трехбайтные, причем после отправки в МК каждого байта необходимо дождаться пачку данных и проанализировать байты B и C (№№ 20 и 21). Первый байт содержит код команды и номер канала, второй и третий байт – аргумент(ы). Сейчас выполняется две команды:

- Изменение Базы
  1 байт = $A0 + N, N – номер канала 0—7.
  2 байт = (База) div $40
  3 байт = (База) mod $40, (База) – значение Базы 0—4095
- Изменение Усиления
  1 байт = $90 + N, N – номер канала 0—7.
  2 байт = (Усил)
  3 байт = (Усил) + $20, (Усил) – значение усиления 0—7