

"Бейсика" БК была использована идея, предложенная П. Чирковым — (см. "Вычислительная техника и ее применение" N 7/91 г.). Аппаратная доработка при этом заключалась в выполнении соединений и установке переключателя SA1 согласно рис.1. При положении "1" переключателя обеспечивается вывод информации на принтер из уже разработанных программ, а в положении "2" — из "Бейсика" БК. Программная часть преобразователя интерфейса "ИРПР" в интерфейс "CENTRONIX" с выполнением перекодирования представлена в таблице. Адрес загрузки &037500.

Работа с данным программно-аппаратным преобразователем интерфейсов заключается в следующем:

BLOAD "EPSPRV" — загрузка,
CLEAR, &037500 — выделение области, используемой "Бейсиком",
DEF USB-&037760,
A-USB(A) — установка приоритета процессора и занесение адреса &037500 в ячейку &0100. Последнюю команду можно запрограммировать.

10046	42700	177400	20027	23	1434	20027	240
2411	10546	162700	240	10705	62705	62	60005
111500	12605	5100	42700	177400	32715	400	1775
10015	52700	1000	10015	32715	400	1375	42700
1000	10015	12600	12716	123014	2	124022	120417
124603	112243	15602	123240	174601	112652	173647	174232
122016	122636	121024	103637	106600	116766	150356	163321
152724	151744	154345	155331	156333	157335	167737	160740
161742	151326	165754	164327	164755	165347	130316	143261
132664	131704	134305	135271	136273	137275	147677	140700
141702	131266	145714	144267	144715	145307	0	0
12737	37500	100	106427	0	207		

Далее можно работать обычным порядком, используя команды LPRINT, LLIST, но не забывая после нажатия клавиши CTOП подавать команду A-USB(A). Переключателями DIP принтера необходимо выбрать кодовую таблицу символов русского языка.

Возможна загрузка драйвера и в стек — с адреса &0400, тогда, естественно, в ячейку &0100 необходимо заносить &0400, а для повторной инициализации драйвера либо использовать команду POKE &0100, &0400, либо переработать соответствующим образом окончание драйвера в каком-либо отладчике.

При использовании предлагаемого программно-аппаратного преобразователя становится возможным использовать почти все возможности принтера "EPSON". Исключение составляет команда DC3, код которой (&H13) "Бейсик" БК использует как возврат курсора на начало строки. Кроме того, отсутствующие в наборе кодов принтера "EPSON" символы "ПИКА", "ТРЕФ", "БУБНЫ", "ЧЕРВИ" и "ПИ" для удобства перекодируются как ESC (&H1B), SO (&H0E), DC4 (&H14), SI (&H0F) и DC2 (&H12) соответственно.

Все сказанное касается, конечно, только импортных принте-

ров, в ПЗУ которых "зашиита" кириллица и советских принтеров с интерфейсом ИРПР-М (МС6313, СМ6337).

Данный программно-аппаратный преобразователь был проверен с принтером EPSON LX-800" и показал хорошие результаты.

Практический совет: в качестве переключателя удобно использовать малогабаритный переключатель диапазонов для транзисторных приемников ПД-2 2П4Н. После небольшого обтачивания с боков он хорошо размещается в корпусе разъема, входящего в комплект БК рядом с плоским кабелем соединения принтера с БК.

П.ЗИМОВЕЦ,
245780, Украина,
г.Конотоп, Сумская обл.,
ул.ген.Тхора, 61 — 99.

"ВЕКТОР" И "РОБОТРОН"

Ниже представлен один из способов реализации последовательного интерфейса C2/RS 232C для ПК типа "Вектор". Автор применил его для связи своего компьютера с принтером "ROBOTRON CM6329.02M".

Интерфейс предельно прост. Выполнен в основном программным путем. В электронной части используется только преобразователь уровней TTL/C2и C2/TTL. За основу взят протокол DTR, требующий всего две линии связи с ПК:

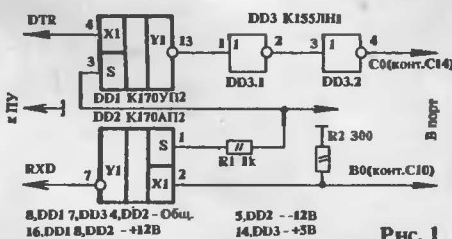


Рис. 1

Табл.1

Скорость передачи (бод)	время на передачу одного бита	
	микросекунды	такты процессора
1200	833.(3)	2088
2400	416.(6)	1043
4800	208.(3)	522
9600	104.1(6)	261

RXD — данные от ПК;

DTR — готовность принтера (ПУ).

После включения ПУ в сеть линия DTR переводится в состояние ВКЛ (сигналом от ПУ). При этом по линии DTR через преобразователь C2/TTL в порт С, а точнее в линию C0 этого порта приходит логический "нуль". Перед очередной передачей данных драйвер проверяет наличие "нуля" в линии C0. При наличии "единицы" программа закидывается в ожидании "нуля". "Единица" означает, что буфер данных ПУ заполнен и дальнейший прием станет возможным только после распечатки его содержимого.

Электрическая схема преобразователя показана на рис.1. Она представляет собой несколько переработанный вариант схемы преобразователя из статьи А.Долгого ("Радио" N6/89 г.). Следует предостеречь от попыток собрать описанный в этой статье транзисторный вариант преобразователя. Опыт автора показал, что желаемого результата при этом достичь не удается. Контакты C14, C18 соответствуют разъему на плате "Вектора". Необходимые уровни напряжения питания — +5В, +12В, -12В можно снять непосредственно с платы ПК.

Рассмотрим технологию разработки драй-

вера. Полагаю, что это будет интересным и полезным для многих начинающих пользователей ПК, которые захотят сами проделать аналогичный путь.

Вначале мной была написана коротенькая программа, формировавшая на выходе одного из портов пользователя меандр (годится любой сигнал). С помощью частотомера затем был определен период T одного колебания. Далее по формуле T/n был рассчитан период времени, соответствующий одному такту (n — количество тактов на один период, суммируется по всем инструкциям, участвующим в формировании одного периода колебания).

В моем ПК длительность такта получилась равной 399 нс, что соответствует тактовой частоте 2,5 МГц.

В табл.1 приведены результаты дальнейших несложных расчетов (для тактовой частоты 2,5 МГц).

Формат передаваемого на ПУ байта должен быть следующим:

1. Стартовый бит (лог. "0");
 2. 7 или 8 бит данных причем 0-й бит передается первым, а 7-й или 8-й последним;
 3. Один или несколько стоповых бит (лог. "1").
- Следует отметить, что 7 первых бит полно-

	DCX	H	
	MVI	A, 091H	
	OUT	04H	;настроить порт
M0:	DI		
	IN	05H	
	ANI	01H	
	JZ	M00	;буфер ПУ занят
M00:	JMP	M0	
	INX	M0	
	MOV	H, A	
	SUB	D	
	JNZ	M01	
	MOV	A, L	
	SUB	E	
	JNZ	M02	
	JMP	EXIT	;выход
M01:	NOP		
	MOV	A, L	
	MOV	A, L	
	MOV	A, L	
M02:	MOV	A, L	
	MOV	A, L	
	CALL	TIM0	
	XRA	B	
	MOV	A, A	
	OUT	06H	;стартовый бит
	MOV	A, M	
	CALL	TIM1	
	OUT	06H	
	MOV	C, A	
	MOV	C, A	
M1:	INR	B	;счетчик бит
	MOV	C, A	
	MOV	A, B	
	CPI	08H	
	JNZ	M2	;8 бит данных
	CALL	TIM3	
	MVI	A, 01H	
	OUT	06H	;1-й стоповый
	CALL	TIM1	
	MVI	A, 01H	
	OUT	06H	;2-й стоповый
	CALL	TIM1	
	MVI	A, 01H	
	OUT	06H	;3-й стоповый
M2:	JMP	M0	
	MOV	A, C	
	RRC		
	CALL	TIM2	
	OUT	06H	
	JMP	M1	
;ЗАДЕРЖКА ВРЕМЕНИ			
TIM1:	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
TIM3:	NOP		
TIM2:	PUSH	B	
	MVI	B, 00H	
	LXI	B, 0000H	
	LXI	B, 0000H	
	POP	B	
TIM0:	PUSH	PSW	
	PUSH	B	
	PUSH	D	
	PUSH	H	
	MVI	A, 00H	
MET:	NOP		
	NOP		
	INR	A	
	CPI	62	
	JNZ	MET	
	POP	H	
	POP	D	
	POP	PSW	
	RET		
EXIT:	EI		
	RST	0	

Табл. 2

стью определяют код символа в соответствии со стандартом КОИ-7. При передаче информации в стандарте КОИ-8 и распечатке дампов используют 8 бит данных.

Какие же рекомендации можно дать на следующем, важнейшем этапе — написа-

нии драйвера? Здесь все во многом определяется квалификацией программиста.

Самое сложное в том, что программа должна быть жестко "привязанной" по времени. Во время побитной передачи данных на ПУ строго регламентирована длительность передачи каждого бита (см. табл.1). В связи с этим каждое, пусть и незначительное, изменение в программе, изменяющее общее число тактов от одного вывода в порт до следующего, влетает за собой скрупулезный подсчет и программную коррекцию команд и тактов процессора. Поэтому не стоит удивляться, увидев в программе автора группы команд, на первый взгляд совершенно бессмысленных. Они служат только для временной задержки.

Для меня остался открытым вопрос, одинаковое ли количество тактов затрачивает процессор, обрабатывая команды условной передачи управления при выполнении и невыполнении условия перехода. К сожалению, мне не удалось это проверить. В имеющейся информации по процессору 580 BM80 такой информации я не нашел. Логически рассуждая, можно прийти к выводу, что при невыполнении условия перехода несколько тактов, во время которых адрес перехода не заносится в память процессора, должны быть потеряны.

Проверить этот вывод проще всего экспериментально, написав специальную программу, выводящую в один из портов поочередно "0" и "1", и замеряя частоту на выходе. Один раз — когда условие перехода не выполняется. Второй раз — при выполнении условия перехода. При этом нужно проследить, чтобы в программу не было внесено других изменений, влияющих на частоту. По разности частот можно было бы определить количество "потерянных" тактов.

Текст драйвера представлен в табл.2 (для работы со скоростью 1200 бод).

Для любителей поэкспериментировать — табл.3.

При переходе к другой скорости передачи никаких изменений в драйвере, кроме TIM0, не требуется. Автор опробовал описываемый драйвер при скорости 9600 бод. Первые 7 бит данных передавались безошибочно. Однако последний (8-й) может быть передан с

Скорость передачи (бод)	время выполнения п/п TИМО (такты процессора)
1200	1962
2400	917
4800	396
9600	135

Табл.3

ошибкой. На скорости 1200 бод этот эффект пропал, а промежуточные режимы не опробывались. Жгут, соединяющий ПК и ПУ не был экранирован. Абсолютная ошибка времени, присутствующая в программе, составляет два такта. На скорости 9600 бодона дает погрешность: $2 \times 100/261 = 0,76\%$. На мень-

ших скоростях она меньше. Драйвер использует системный стек, но это не обязательно. Стек можно организовать в любом месте ОЗУ. Если драйвер будет загружаться с адреса 40Н, то вершину стека удобно поместить по адресу 100Н. Автору известны три текстовых редактора для ПК "Вектор". Это "СТАРТ-1200", "EDACM", "RETEX". О "RETEX"е разговор особый — если эта тема заинтересует пользователей, то автор может предоставить материал о включении драйвера в состав этого редактора.

С "EDASM"ом никаких проблем: загружаете в HL 2800H (адрес начала текстового буфера), в DE — адрес конца буфера и передаете управление драйверу. Буфер распечатывается на ПУ. Следует помнить, что вершина стека располагается по адресу 5000 H. Если текст длинный, то директивой "X" следует изменить содержимое SP, а затем выйти в редактор и загрузить текст.

Сложнее с распечаткой буфера "СТАРТ-1200". Для экономии ОЗУ этот редактор по своему кодирует некоторые символы. Кроме того, он не заполняет буфер идущими подряд пробелами или знаками табуляции, а суммирует их и в буфер выдает уже сумму. В "Мониторе" редактора "СТАРТ-1200" имеются средства для расшивки и распечатки такого буфера (п/п 7341Н — выдача текста на принтер), но эта подпрограмма рассчитана на параллельный интерфейс "CENTRONIX". Лучшее, что можно в этом случае придумать, — это расшивать буфер перед каждой посылкой строки на ПУ. Однако такой драйвер еще предстоит написать. Второй выход — это написать еще одну подпрограмму, преобразовывающую стандарт буфера "СТАРТ-1200" в стандарт ASCII. Досконально этим вопросом автор пока не занимался. Программа, приведенная в табл.4, написана и опробована в сжатые сроки. Пробелы, идущие в строке подряд (от 2 до 9), интерпретируются, как один знак табуляции.

Табл.4

TO:	DI	C.L
TOT:	MOV	H
	INX	A.H
	MOV	D
	SUB	T01
	INJ	A.L
	MOV	E
	SUB	T02
T01:	JZ	A.M
	MOV	A
	ANA	CR
	JZ	01H
	CPI	SP
	JZ	09H
	CPI	HT
	JC	T0
	JMP	A.09H
HT:	MVI	M.A
HT1:	MOV	T0
	JMP	A.20H
SP:	MVI	HT1
	JMP	A.L
CR:	MOV	C
	SUB	04H
	CPI	T02
	JZ	A.0DH
	MVI	M.A
	MOV	T01
	JMP	
	BI	0
T02:	EST	