

<u>PS</u>	<u>M. Dili</u>	<u>Simgesel Dil</u>			<u>Açıklama</u>
0000	BD 00 16	BASLA	JSR	KOSUL	
0003	86 EA		LDAA	#\$EA	
0005	B7 10 05		STAA	\$1005	
0008	B6 10 05		LDAA	\$1005	
000B	BD 00 24		JSR	GONDER	Gönderme işlemi
000E	4F		CLRA		
000F	BD 00 2F		JSR	AL	Alma işlemi
0012	B7 10 06		STAA	\$1006	
0015	3F		SWI		
0016					
0016	7F 88 01	KOSUL	CLR	\$8801	KOSUL alt programı
0019	86 98		LDAA	#\$98	
001B	B7 88 03		STAA	\$8803	Denetim kütüğü koşullanıyor
001E	86 45		LDAA	#\$45	
0020	B7 88 02		STAA	\$8802	Komut kütüğü koşullanıyor
0023	39		RTS		
0024					
0024	F6 88 01	GONDER	LDAB	\$8801	GONDER alt programı
0027	C4 10		ANDB	#\$10	Verici boş mu?
0029	27 F9		BEQ	GONDER	Boş değil ise tekrar kontrol et
002B	B7 88 00		STAA	\$8800	Boş ise veriyi iskeleye yaz
002E	39		RTS		
002F					
002F	F6 88 01	AL	LDAB	\$8801	AL alt programı
0032	C4 08		ANDB	#\$08	Alıcı dolu mu?
0034	27 F9		BEQ	AL	Dolu değil ise tekrar kontrol et
0036	B6 88 00		LDAA	\$8800	Dolu ise veriyi iskeleden al
0039	39		RTS		

<u>PS</u>	<u>M. Dili</u>	<u>Simgesel Dil</u>			<u>Açıklama</u>
0000	BD 00 2A	BASLA	JSR	KOSUL	ASIA'nın koşullanması
0003	CE 44 00		LDX	#\$4400	Sayı kümesinin başlangıç adresi
0006	FF 10 00		STX	\$1000	
0009	CE 45 00		LDX	#\$4500	Yazmaya başlanacak bellek gözü
000C	FF 10 02	GERI	STX	\$1002	
000F	FE 10 00		LDX	\$1000	
0012	A6 00		LDAA	0,X	
0014	BD 00 38		JSR	GONDER	Bir veri gönder
0017	08		INX		
0018	FF 10 00		STX	\$1000	SK'yi sakla
001B	FE 10 02		LDX	\$1002	Yazılacak yerin bellek adresini al
001E	BD 00 43		JSR	AL	Gönderilen veriyi al
0021	A7 00		STAA	0,X	Sıradaki bellek adresine yaz
0023	08		INX		
0024	8C 45 10		CPX	#\$4510	Sona geldik mi?
0027	26 E3		BNE	GERI	Gelmediyse devam et
0029	3F		SWI		
002A					
002A	7F 88 01	KOSUL	CLR	\$8801	KOSUL alt programı
002D	86 98		LDAA	#\$98	
002F	B7 88 03		STAA	\$8803	
0032	86 45		LDAA	#\$45	
0034	B7 88 02		STAA	\$8802	
0037	39		RTS		
0038					
0038	F6 88 01	GONDER	LDAB	\$8801	GONDER alt programı
003B	C4 10		ANDB	#\$10	
003D	27 F9		BEQ	GONDER	
003F	B7 88 00		STAA	\$8800	
0042	39		RTS		

```

0043
0043 F6 88 01 AL LDAB $8801 AL alt programı
0046 C4 08 ANDB #$08
0048 27 F9 BEQ AL
004A B6 88 00 LDAA $8800
004D 39 RTS

```

Temel ASIA Deneyi

ASIA'nın durum kütüğü ve komut kütüğü,

İletişim hızı : 1200bps

Veri boyu : 8bit

Dur biti : 2bit

Eşlik : Çift eşlik

R6551'in istenen biçimde çalışabilmesi için, iç kütüklerinin sıfırlanması gerekmektedir.

Buna göre şu şekilde koşullamalar yapılmalıdır:

Durum → %1001 1000 = \$98

Komut → %0100 0101 = \$45

Program:

PS	M. Dili	Simgesel Dil	Açıklama
0000	BD 00 16	BASLA JSR KOSUL	
0003	86 EA	LDAA #\$EA	
0005	B7 40 05	STAA \$4005	
0008	B6 40 05	LDAA \$4005	
000B	BD 00 24	JSR GONDER	Gönderme işlemi
000E	4F	CLRA	
000F	BD 00 2F	JSR AL	Alma işlemi
0012	B7 40 06	STAA \$4006	
0015	3F	SWI	
0016			
0016	7F 88 01	KOSUL CLR \$8801	KOSUL alt programı
0019	86 98	LDAA #\$98	
001B	B7 88 03	STAA \$8803	Denetim kütüğü koşullanıyor
001E	86 45	LDAA #\$45	
0020	B7 88 02	STAA \$8802	Komut kütüğü koşullanıyor
0023	39	RTS	
0024			
0024	F6 88 01	GONDER LDAB \$8801	GONDER alt programı
0027	C4 10	ANDB #\$10	Verici boş mu?
0029	27 F9	BEQ GONDER	Boş değil ise tekrar kontrol et
002B	B7 88 00	STAA \$8800	Boş ise veriyi iskeleye yaz
002E	39	RTS	
002F			
002F	F6 88 01	AL LDAB \$8801	AL alt programı
0032	C4 08	ANDB #\$08	Alıcı dolu mu?
0034	27 F9	BEQ AL	Dolu değil ise tekrar kontrol et
0036	B6 88 00	LDAA \$8800	Dolu ise veriyi iskeleden al
0039	39	RTS	

Program çalıştırılmadan önce R6551 kırımının verici çıkışı (TxD), alıcı girişine (RxD); gönderme isteği çıkışı (RTS), gönderme için sil girişine (CTS) bağlanmıştır. Ayrıca DCD ve DSR girişleri toprağa bağlanmıştır. Bu bağlantıyı yapmanın amacı ASIA'nın gönderdiği veriyi tekrar kendisinin almasını sağlamaktır.

Program çalıştırılmadan önce \$4005 ve \$4006 sayılı bellek gözlerine \$00 yüklenmiş ve program çalıştırılmıştır. Bunun amacı bu bellek gözlerinde herhangi bir değişme olduğunda sağlıklı bir şekilde gözlemleyebilmektir. Program çalıştırdıktan sonra \$4005 ve \$4006 bellek gözlerinde \$EA değerinin yazılı olduğu gözlenmiştir. Bu işlemi bir çok defa yinelediğimizde gene aynı sonuçlarla karşılaşmıştık. Son olarak da programımızın sağlıklı bir şekilde çalışıp çalışmadığını kontrol etmek amacıyla ilk başta yapmış olduğumuz (TxD – RxD &&RTS– CTS&&DCD -- DSR) bağlantılarını sağlayan kablolar çıkartılmış ve program bu halde yeniden çalıştırılmıştır. Bu durumda ise \$4005 sayılı bellek gözüne \$EA değeri yazılıyorken \$4006 sayılı bellek gözünde ise herhangi bir değişme gözlenmemiştir.

ASIA Üzerinden Bir Küme Veri Gönderme ve Alma Deneyi

Program:

<u>PS</u>	<u>M. Dili</u>	<u>Simgesel Dil</u>	<u>Açıklama</u>
0000	BD 00 2A	BASLA JSR KOSUL	ASIA'nın koşullanması
0003	CE 44 00	LDX #\$4400	Sayı kümesinin başlangıç adresi
0006	FF 40 00	STX \$4000	
0009	CE 45 00	LDX #\$4500	Yazmaya başlanacak bellek gözü
000C	FF 40 02	GERI STX \$4002	
000F	FE 40 00	LDX \$4000	
0012	A6 00	LDAA 0,X	
0014	BD 00 38	JSR GONDER	Bir veri gönder
0017	08	INX	
0018	FF 40 00	STX \$4000	SK'yı sakla
001B	FE 40 02	LDX \$4002	Yazılacak yerin bellek adresini al
001E	BD 00 43	JSR AL	Gönderilen veriyi al
0021	A7 00	STAA 0,X	Sıradaki bellek adresine yaz
0023	08	INX	
0024	8C 45 10	CPX #\$4510	Sona geldik mi?
0027	26 E3	BNE GERI	Gelmediyse devam et
0029	3F	SWI	
002A			
002A	7F 88 01	KOSUL CLR \$8801	KOSUL alt programı
002D	86 98	LDAA #\$98	
002F	B7 88 03	STAA \$8803	
0032	86 45	LDAA #\$45	
0034	B7 88 02	STAA \$8802	
0037	39	RTS	
0038			
0038	F6 88 01	GONDER LDAB \$8801	GONDER alt programı
003B	C4 10	ANDB #\$10	
003D	27 F9	BEQ GONDER	
003F	B7 88 00	STAA \$8800	
0042	39	RTS	
0043			
0043	F6 88 01	AL LDAB \$8801	AL alt programı
0046	C4 08	ANDB #\$08	
0048	27 F9	BEQ AL	
004A	B6 88 00	LDAA \$8800	
004D	39	RTS	

ASIA'yı Temel ASIA Deneyindeki belirtilen ayarlara uygun olacak şekilde tekrar koşulladık. İlk deneydeki gerçekleştirilen bağlantıları aynen koruduk.

Durum → %1001 1000 = \$98

Komut → %0100 0101 = \$45

Program çalıştırılmadan önce bellekte \$4400 sayılı bellek gözünden başlamak üzere 16 değişik değer yazdık.

Değerler;

\$00, \$01, \$02, \$03, \$04, \$05, \$06, \$07, \$08, \$09, \$0A, \$0B, \$0C, \$0D, \$0E, \$0F

\$4400 sayılı bellek gözünden başlayan bu verileri ASIA vericisi üzerinden asenkron seri olarak bilgisayar dışına gönderecek verici ve \$4500 sayılı bellek gözüne bu verileri okuyup yazacak alıcı program olan yukarıdaki programımızı \$0000 sayılı bellek gözünden başlayarak yazdık ve çalıştırdık. Sonuçta \$4400 sayılı bellek gözünden başlayarak yazmış olduğumuz veriler sırasıyla \$4500 sayılı bellek gözüne yazılmış olarak gözlemledik.

1. Temel ASIA Deneyi İçin Yazılan Program:

ASIA'nın durum kütüğü ve komut kütüğü,

İletişim hızı : 1200bps
Veri boyu : 8bit
Dur biti : 2bit
Eşlik : Çift eşlik

koşullarına göre şu şekilde olmalıdır:

Durum → %1001 1000 = \$98

Komut → %0100 0101 = \$45

<u>PS</u>	<u>M. Dili</u>	<u>Simgesel Dil</u>	<u>Açıklama</u>
0000	BD 00 16	BASLA JSR KOSUL	
0003	86 EA	LDAA #\$EA	
0005	B7 10 05	STAA \$1005	
0008	B6 10 05	LDAA \$1005	
000B	BD 00 24	JSR GONDER	Gönderme işlemi
000E	4F	CLRA	
000F	BD 00 2F	JSR AL	Alma işlemi
0012	B7 10 06	STAA \$1006	
0015	3F	SWI	
0016			
0016	7F 88 01	KOSUL CLR \$8801	KOSUL alt programı
0019	86 98	LDAA #\$98	
001B	B7 88 03	STAA \$8803	Denetim kütüğü koşullanıyor
001E	86 45	LDAA #\$45	
0020	B7 88 02	STAA \$8802	Komut kütüğü koşullanıyor
0023	39	RTS	
0024			
0024	F6 88 01	GONDER LDAB \$8801	GONDER alt programı
0027	C4 10	ANDB #\$10	Verici boş mu?
0029	27 F9	BEQ GONDER	Boş değil ise tekrar kontrol et
002B	B7 88 00	STAA \$8800	Boş ise veriyi iskeleye yaz
002E	39	RTS	
002F			
002F	F6 88 01	AL LDAB \$8801	AL alt programı
0032	C4 08	ANDB #\$08	Alıcı dolu mu?
0034	27 F9	BEQ AL	Dolu değil ise tekrar kontrol et
0036	B6 88 00	LDAA \$8800	Dolu ise veriyi iskeleden al
0039	39	RTS	

Program çalıştırılmadan önce R6551 kırmığının verici çıkışı (TxD), alıcı girişine (RxD); gönderme isteği çıkışı (RTS), gönderme için sil girişine (CTS) bağlanmıştır. Bu bağlantıyı yapmanın amacı ASIA'nın gönderdiği veriyi tekrar kendisinin almasını sağlamaktır.

Program çalıştırılmadan önce \$1005 ve \$1006 sayılı bellek gözlerine \$00 yüklenmiş ve program çalıştırılmıştır. Sonrasında bu bellek gözlerine bakıldığında \$1005'e program tarafından \$EA değeri, \$1006'ya ise iletilen \$EA değeri yazılmıştır. Devrenin çalışıp çalışmadığını kontrol amacıyla bağlantıyı sağlayan kablolar çıkarılıp program tekrar çalıştırılmış ve bu sefer herhangi bir veri iletimi olmadığından \$1006 sayılı bellek gözüne rasgele bir sayı yazıldığı görülmüştür.

2. ASIA Üzerinden Bir Küme Veri Gönderme ve Alma Deneyi İçin Yazılan Program:

PS	M. Dili	Simgesel Dil	Açıklama
0000	BD 00 2A	BASLA JSR KOSUL	ASIA'nın koşullanması
0003	CE 44 00	LDX #\$4400	Sayı kümesinin başlangıç adresi
0006	FF 10 00	STX \$1000	
0009	CE 45 00	LDX #\$4500	Yazmaya başlanacak bellek gözü
000C	FF 10 02	GERI STX \$1002	
000F	FE 10 00	LDX \$1000	
0012	A6 00	LDAA 0,X	
0014	BD 00 38	JSR GONDER	Bir veri gönder
0017	08	INX	
0018	FF 10 00	STX \$1000	SK'yı sakla
001B	FE 10 02	LDX \$1002	Yazılacak yerin bellek adresini al
001E	BD 00 43	JSR AL	Gönderilen veriyi al
0021	A7 00	STAA 0,X	Sıradaki bellek adresine yaz
0023	08	INX	
0024	8C 45 10	CPX #\$4510	Sona geldik mi?
0027	26 E3	BNE GERI	Gelmediyse devam et
0029	3F	SWI	
002A			
002A	7F 88 01	KOSUL CLR \$8801	KOSUL alt programı
002D	86 98	LDAA #\$98	
002F	B7 88 03	STAA \$8803	
0032	86 45	LDAA #\$45	
0034	B7 88 02	STAA \$8802	
0037	39	RTS	
0038			
0038	F6 88 01	GONDER LDAB \$8801	GONDER alt programı
003B	C4 10	ANDB #\$10	
003D	27 F9	BEQ GONDER	
003F	B7 88 00	STAA \$8800	
0042	39	RTS	
0043			
0043	F6 88 01	AL LDAB \$8801	AL alt programı
0046	C4 08	ANDB #\$08	
0048	27 F9	BEQ AL	
004A	B6 88 00	LDAA \$8800	
004D	39	RTS	

Program ilk sayıdan başlayarak önce gönderme sonra alma işlemlerini gerçekleştirerek sırası ile sayıları iletmektedir. Daha önce kurulan mevcut bağlantılar korunarak program çalıştırılmış ve istenilen 16 sayı iletilerek yeni bellek gözlerine yazıldığı görülmüştür.