

### Soru-3 (30 P)

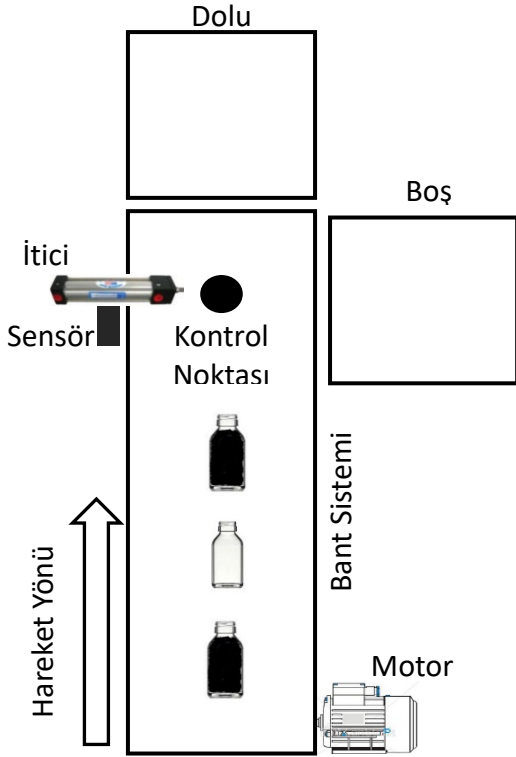
Bir GPS (Küresel konumlama sistemi) cihazı ile bir mikrodenetleyici haberleştirilmektedir. GPS alıcısının her 250 us'de (mikrosaniye) bir 3 byte'lık bir paketi mikrodenetleyicinin P1 portuna gönderdiği varsayılmaktadır. Dolayısı ile mikrodenetleyicinin de her 250 us'de bir, P1 portunu okuduğu kabul edilecektir. Program, P1 portunu okuduktan sonra sonsuz döngülü ve boş **bekle** altprogramına dallanmaktadır.

**GPS alıcısının veri gönderim kuralı şu şekildedir:**

Art arda gönderilen 3 baytın ilk ikisi, verinin türünü, son bayt ise veriyi gösterir. Örneğin; 414CFF (41h='A', 4Ch='L', FFh=255m (yüksekliğin verisi)). "AL" karakterleri yükseltinin gönderildiğini belirtmektedir. Eğer bunların dışında bir karakter dizisi gelirse mikrodenetleyici bu mesajı dikkate almaz. Verilen bilgilere göre, P1 portundan okunan veriler **yükseklik verisi** ise bu yükseklik verisini 5000h adresine yazan programı 8051 assembly dilinde **T0 ile kesme tabanlı** olarak tasarlayınız. İşlemcinin her makine çevrimi 1mikrosaniyedir (MC=1us).

**Önemli NOT: SFR atamalarını BAYT ve HEX tabanlı olarak yapınız**

### Soru-4 (35 P)



**İtici, P1.0 pinine;**

**Sensor, harici kesme INT0'a;**

**Motor, P1.1'e bağlıdır...**

Sistem şu şekilde çalışmaktadır:

- Sistem ilk başladığında motor =1 (bantı yürüt), itici=0 (dur) olmalıdır
- Kontrol noktasına gelen şişe dolu ise herhangi bir eylem yapılmayacaktır. Boş bir şişe kontrol noktasına erişirse normalde lojik-1 üreten sensor, düşen bir kenar üretilir tekrar lojik -1'e dönmektedir. Bu durumda, bant durdurulur ve itici 60ms boyunca lojik-1 yapılır ve boş şişe "Boş" isimli sepete itilir. 60ms sonunda itici lojik-0 yapılarak ilk konumuna ayarlanır ve bant tekrar yürütülmeye başlanır.

Yukarıdaki senaryoyu 1us makine çevrimine sahip bir mikrodenetleyici kullanarak kesme ve T1 zamanlayıcısını kullanarak tasarlayınız.

**Önemli NOT: SFR atamalarını BAYT ve HEX tabanlı olarak yapınız**

TCON Zamanlayıcı/Sayıcı Kontrol Saklayıcısı	
88H	0 0 0 0 0 0 0 0 Reset Değeri
TF 1	TR 1
TF 0	TR 0
IE 1	IT 1
IE 0	IT 0
Bit 7	Bit 0

IE Kesme Yetkilendirme Saklayıcısı	
A8H	0 0 0 0 0 0 0 0 Reset Değeri
EA	ET2
ES	ET1
EX1	ET0
EX0	
Bit 7	Bit 0

TMOD Zamanlayıcı MOD Seçim Saklayıcısı	
89H	0 0 0 0 0 0 0 0 Reset Değeri
GATE	C/T
M1	M0
GATE	C/T
M1	M0
Bit 7	Bit 0
T1	T0

IP Kesme Önceliği Kontrol Saklayıcısı	
B8H	0 0 0 0 0 0 0 0 Reset Değeri
PT2	PS
PT1	PX1
PT0	PX0
Bit 7	Bit 0

## Soru-1 (Her şık 2p, toplam 20p)

Aşağıdaki boşlukları doldurunuz

- Z/S biriminin saat darbesi harici olarak uygulanıyorsa Z/S birimi **Counter/Sayıci** modunda çalıştırılmalıdır.
- 8051 **8** bitlik bir denetleyicidir.
- Reset vektörü haric tüm kesmeler için **8.bayt** yer ayrılmıştır.
- Kesme öncelikleri **.IP** .kaydedicisi ile belirlenir
- Z/S konfigürasyon (ayar) kaydedicileri **SFR** belleğinde bulunur
- Harici kesme birimlerinin düşen kenarda etkin olmaları için **IT0/IT1** bitleri kullanılır
- SJMP ile geri yönde **128.bayt** ve ileri yönde **127.bayt** dallanabilir
- e) AJMP komutu **2** bayt yer kaplar
- f) Kesme rutininden dönerken **RETI** .komutu kullanılmalıdır
- Bir kesme oluştuğunda bir sonraki adres değeri **.STACK/ YİĞİN** .belleğe kaydedilir

## Soru-2 (5p+5p+5p)

```
ORG 0h
MOV R6, #1Ah
MOV R7, #44h
MOV R4, #22h
MOV R5, #0DBh
CALL SUB
JMP $
SUB: MOV A, R7
ADD A, R5
MOV R3, A
MOV A, R6
ADDC A, R4
MOV R2, A
MOV A, #00h
ADDC A, #00h
MOV R1, A
RET
```

a)Yandaki programın amacı nedir? (5p)

**Program 16-bit toplama yapmaktadır.**

b)Program sonsuz döngüye girdikten sonra R2 ve R3'ün değerleri nedir?(5p+5p)

R2=.....**03Dh**

R3=...**1Fh**

### CEVAP-4 (35P)

10 PUAN	<b>ORG 0H</b>	;RESET VEKTÖRÜ
	<b>JMP MAIN</b>	;ANA PROGRAMA GİT
	<b>ORG 03H</b>	;INTO KESME VEKTÖRÜ
	<b>MOV P1,#01</b>	; İTİCİ=1, MOTOR=0
	;60ms=60000 SAYMA, 65536-60000=5536	
	<b>MOV TL1,#LOW(5536)</b>	; DÜŞÜK BAYT
	<b>MOV TH1,#HIGH(5536)</b>	; YÜKSEK BAYT
	<b>MOV TCON,#41H</b>	;TIMER1 RUN, IT0=1
	<b>RETI</b>	
10 PUAN	<b>ORG 1BH</b>	;T1 KESME VEKTÖRÜ
	<b>MOV P1,#01</b>	; İTİCİ=0, MOTOR=0
	<b>MOV TCON,#0H</b>	;TIMER1 STOP
	<b>RETI</b>	
15 PUAN	<b>ORG 30H</b>	;ANA BAŞLANGIÇ
MAIN:	<b>MOV P1,#02</b>	;İTİCİ=0, MOTOR=1
	<b>MOV IE,#89H</b>	;SETB EX0, ET1, EA
	<b>MOV TMOD,#10H</b>	;T1- 16-BİT TİMER
	<b>JMP \$</b>	;SONSUZ DÖNGÜ
	<b>END</b>	

### CEVAP-3 (30p)

5P		ORG 0H	;RESET VEKTÖRÜ
		JMP MAIN	;ANA PROGRAMA DALLAN
		ORG 0BH	;TO KESME VEKTÖRÜ
		JMP TO_ISR	;TO_ISR DALLAN
10P	MAIN:	ORG 30H	;ANA PROGRAM BAŞLANGICI
		MOV IE,#82H	;EA=1, ET0=1
		MOV TMOD,#02H	;T0=8-OTOMATİK YÜKLEME
		MOV TL0,#6	; 256-250=6 DAN DOLAYI
		MOV TH0,TL0	; TH0 OTO YÜKLEME DEĞERİ
		MOV TCON,#10H	;SETB TRO KABUL DEĞİL, SORU ŞARTI
	BEKLE:	MOV DPTR,#5000	;XRAM BAŞLANGIÇ ADRESİ
		CLR A	;AKÜ=0, KARAKTER SAYICI
	JMP \$	;SONSUZ BOŞ DÖNGÜ	
15P	TO_ISR:	CJNE A,#2,KOMT	;AKÜ≠2 DEĞİLSE OKUMAYA DEVAM
		MOV A,P1	;AKÜ=2 İSE VERİYİ AKÜYE KAYDET
		MOVX @DPTR,A	;VERİYİ 5000 ADRESİNE YAZ
		JMP CLR_A	;AKÜYÜ SIFIRLA, YENİ VERİ BEKLE
	KOMT:	JZ ILK_CH	;AKÜ=0 İSE İLK KARAKTERİ OKU
		MOV R1,P1	;DEĞİLSE İKİNCİ KARAKTERİ OKU
	ILK_CH:	CJNE R1,#4CH,CLR_A	;‘L’ KARAKTERİ DEĞİLSE RESETLE
		INC A	;OKUNAN ‘L’, AKÜ=2
		RETI	; ANA PROGRAMA DÖN
		MOV R0,P1	; İLK CHAR OKUNDU
	CLR_A:	CJNE R0,#41H,CLR_A	;‘A’ KARAKTERİ DEĞİLSE RESETLE
		INC A	; OKUNAN CHAR=‘A’, AKÜ=1
RETI		; ANA PROGRAMA DÖN	
CLR A		; ‘A’ DEĞİL O ZAMAN SIFIRLA	
	RETI	; ANA PROGRAMA DÖN	
	END		