***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.

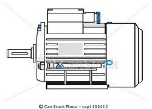
**GPS alıcısının veri gönderim kuralı şu şekildedir:**

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;**



Motor

Boş

Dolu

Bant Sistemi



İtici

Hareket Yönü

Sensör

Kontrol Noktası



**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 üretip 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**



**

***Soru-1 (Her şık 2p, toplam 20p)***

**Aşağıdaki boşlukları doldurunuz**

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

10 PUAN

10 PUAN

15 PUAN

***CEVAP-4 (35P)***

***ORG 0H*** *;RESET VEKTÖRÜ*

***JMP MAIN*** *;ANA PROGRAMA GİT*

***ORG 03H*** *;INT0 KESME VEKTÖRÜ*

***MOV P1,#01*** *; İTİCİ=1, MOTOR=0*

*;60ms=60000 SAYMA, 65536-60000=5536*

***MOV TL1,#LOW(5536)*** *; DÜŞÜK BAYT*

***MOV TH1,#HIGH(5536)*** *; YÜKSEK BAYT*

***MOV TCON,#41H*** *;TIMER1 RUN, IT0=1*

***RETI***

***ORG 1BH*** *;T1 KESME VEKTÖRÜ*

***MOV P1,#01*** *; İTİCİ=0, MOTOR=0*

***MOV TCON,#0H*** *;TIMER1 STOP*

***RETI***

***ORG 30H*** *;ANA BAŞLANGIÇ*

*MAIN:* ***MOV P1,#02*** *;İTİCİ=0, MOTOR=1*

***MOV IE,#89H*** *;SETB EX0, ET1, EA*

***MOV TMOD,#10H*** *;T1- 16-BİT TİMER*

***JMP $*** *;SONSUZ DÖNGÜ*

***END***

5P

10P

15P

**CEVAP-3 (30p)**

**ORG 0H** ;RESET VEKTÖRÜ

**JMP MAIN** ;ANA PROGRAMA DALLAN

**ORG 0BH** ;T0 KESME VEKTÖRÜ

**JMP T0\_ISR ;**T0\_ISR DALLAN

**ORG 30H** ;ANA PROGRAM BAŞLANGICI

MAIN: **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 TR0 KABUL DEĞİL, SORU ŞARTI

**MOV DPTR,#5000** ;XRAM BAŞLANGIÇ ADRESİ

**CLR A** ;AKÜ=0, KARAKTER SAYICI

BEKLE: **JMP $** ;SONSUZ BOŞ DÖNGÜ

T0\_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 IKINCI KARAKTERI OKU

**CJNE R1,#4CH,CLR\_A** ;’L’ KARAKTERİ DEĞİLSE RESETLE

**INC A** ;OKUNAN ‘L’, AKÜ=2

**RETI** ; ANA PROGRAMA DÖN

ILK\_CH: **MOV R0,P1** ; İLK CHAR OKUNDU

**CJNE R0,#41H,CLR\_A** ;’A’ KARAKTERİ DEĞİLSE RESETLE

**INC A** ; OKUNAN CHAR=’A’, AKÜ=1

**RETI** ; ANA PROGRAMA DÖN

CLR\_A: **CLR A** ; ‘A’ DEĞİL O ZAMAN SIFIRLA

**RETI** ; ANA PROGRAMA DÖN

**END**