

Binary ASK

1) $F_c=100\text{Hz}$ olarak belirledik

2) $F_s=1100\text{Hz}$ olarak belirledim. $\implies (F_s > F_c)$

3) $100 \cdot \sin(2 \cdot \pi \cdot f_s / f_c \cdot t)$

Başlangıç değerleri:

$Y[0] = 100 \cdot \cos(2 \cdot 3,14 \cdot (1100) \cdot 1) = 31$

$Y[1] = 100 \cdot \cos(2 \cdot 3,14 \cdot (1100) \cdot 2) = -79$

Timer interrupt:

TMOD = 0x01 \rightarrow Timer 0 mod 1 olarak ayarladık.

SCON = 0x50 \rightarrow 8 Data biti 1 başlangıç biti ,1 stop biti

TH0 = 0xFC \rightarrow FFFF-FCBA=897 \rightarrow 897+1=898 \rightarrow 898*1,085us=909ms=1/Fs=1/1100=909ms=Ts

TLO = 0xBA

TR0 = 1 \rightarrow Timer0 enable yapar.

TH1 = 0xFD \rightarrow 9600 baudrate ayarlar.

TR1 = 1 \rightarrow Timer1 enable yapar.

ES = 1; \rightarrow Enable Serial interrupt

ET0 = 1; \rightarrow Enable Timer0 Interrupt

EA = 1; \rightarrow Enable Global Interrupt bit

PS = 1 \rightarrow Serial port interrupt öncelik seviyesini high programlar.

4) Serial porttan data alma:

receivedChar = SBUF; \rightarrow Sbuf ı receivedChara atadım.

SBUF = receivedChar;

RI = 0;

charAvailable=1;

Daha sonra gelen karakteri ascii olarak bitlere ayırdık.Bit eğer 1 is önce low sonra high eğer 0 ise önce high sonra low gönderdik.



KOD:

```
#include<reg51.h>          /* Include x51 header file */

int y[2] = {31,-79}; // IIR filtresi baslangic degerlerini atayin.

                                // y[0] simdiki, y[1] bir onceki zamandaki degerler

int M[2][2] = {{15,-8},{1,0}};          // M matrisi ile carpinca birim zaman sonraki
degerler elde edilir.

bit charAvailable = 0; // Gonderilecek veri oldugunu bildirir.

bit currentHalf = 0; // Yuksek oldugu zaman P1 den sinus gondermeye devam edilir.

bit bitValue = 0;          // Simdi gonderilen bit degerini tutar.

bit idle = 1;              // Data gonderildigi surece 0 olur.

int a = 0;                 // Sadece ara degisken

int halfCounter = 0; // 1 oldugu zaman sinus gonderiliyordur.

unsigned char receivedChar=0; // Seri porttan alinan deger

unsigned char counter=0;    //Sadece counter.

void IIR_set(void)
{
    y[0] = 31;          // Sinus baslangic degerleri ata.
    y[1] = -79;
}

void IIR_reset(void)
{
    y[0] = 0;
    y[1] = 0;
}

void sendHigh(void)
{
    IIR_set();
    currentHalf=1; // Sinus gonderilecek
    halfCounter=100 ; // Kac ornekleme zamani(Ts) kadar sinus gonderecek?
}

void sendLow(void)
{
    IIR_reset();
```

```

    currentHalf=0; // Sinus kesilecek

    halfCounter=100; // Kac ornekleme zamani(Ts) kadar sinus sinyalinini kesecek?
}

void sendZero(void)
{
    sendHigh();// Manchester kodlamaya gore hangi fonksiyon olduguna karar ver.
    while(halfCounter);
    sendLow();// Manchester kodlamaya gore hangi fonksiyon olduguna karar ver.
    while(halfCounter);
}

void sendOne(void)
{
    sendLow();// Manchester kodlamaya gore hangi fonksiyon olduguna karar ver.
    while(halfCounter);
    sendHigh();// Manchester kodlamaya gore hangi fonksiyon olduguna karar ver.
    while(halfCounter);
}

unsigned char IIR(void)
{
    a=y[1];
    y[1]=y[0];
    y[0]=M[0][0]*y[0]+M[0][1]*a;
    y[0]=(y[0]>>3); //IIR filtresi cikisini(y[0]) guncelle.

    return (unsigned char)(y[0] + 128); // DC degeri 128e cektik ve veri tipini 1byte olarak degistirdik.
}

void Timer_Serial_init()
{
    TMOD = 0x01; // Timerlar ve serial comm. kontrolu
    SCON = 0x50;
    TH0 = 0xFC;
    TL0 = 0xBA;
    TR0 = 1;
    TH1 = 0xFD; // Load timer 1 value for 9600 baudrate
    TR1 = 1;

```

```

    ES = 1;    // Enable Serial INterrupt
    ET0 = 1;  //Enable TImeR0 Interrupt
    EA = 1;    //Enable Global Interrupt bit

    PS = 1;

    P2 = 0;
    P1 = 0;
}

void Timer0_ISR() interrupt 1      /* Timer0 interrupt service routine (ISR) */
{
    if(idle){
        P2=128;// Portlari guncelle
    }
    else {
        halfCounter--;
        P2=IIR();                /* Portlari, timer0 i ve halfCounteri guncelle*/
    }
    TH0=0xFC;
    TL0=0xBA ;
    P1=~P1;
}

void serial_isr() interrupt 4
{
    if(RI == 1)
    {
        receivedChar = SBUF;
        SBUF = receivedChar;
        RI = 0;

        charAvailable=1;
    }
    else if(TI == 1)

```

```

{
    TI=0;
}
}

void sendCharacter(void){
    idle = 0;
    sendOne(); // Baslangic biti 1 olacak (Manchester coding).
    for( counter=8; counter>0; counter--){
        bitValue = (receivedChar >> (counter-1)) & 1; // Siradaki bit degerini alir.
        if(bitValue==1){
            sendOne();
        }else{sendZero();
        }
    }
    idle = 1;
}

void main(void)
{

    Timer_Serial_init();
    IIR_reset();

    while(1){

        if(charAvailable){
            charAvailable=0;
            sendCharacter();
        }

    }
}

```