

Task program

PROGRAM:-TRANSMITTER(TX):-

```

#include <xc.h>
#define _XTAL_FREQ 6000000 //initialize the clock speed
void init(void); //function declaration
void lcdcmd(unsigned char); //function declaration
void lcddata(unsigned char); //function declaration
unsigned char i,val,rxval,j,tx[10]="TX DATA=",rx[10]="RX DATA="; //declare the variable
void transmit_init(); //function declaration

void main()
] {
    init(); //call the init function
    lcdcmd(0x80); //set the location in lcd1
    for(i=0;i!=8;i++)
        lcddata(tx[i]); //sent the tx data to print in the lcd
    lcdcmd(0xC0); //set the location of the lcd
    for(i=0;i!=8;i++)
        lcddata(rx[i]); //sent the rx data to print in the lcd
    while(1)
    {
        val=PORTB; //port b data will be sent to the val variable
        lcdcmd(0x88); //set the location of the lcd
        switch(val) //val will be given to the switch case
        {
            case 0xE0: //case1 1110 0000
                TXREG='A'; //capital A will be sent to the TXREG register
                lcddata('A'); //sent the A to the lcd
                break;
            case 0xD0: //case2 1101 0000
                TXREG='B'; //capital B will be sent to the TXREG register
                lcddata('B'); //sent the B to the lcd
                break;
            case 0xB0: //case3 1011 0000
                TXREG='C'; //capital C will be sent to the TXREG register
                lcddata('C'); //sent the C to the lcd
                break;
            case 0x70: //case4 0111 0000
                TXREG='D'; //capital D will be sent to the TXREG register
                lcddata('D'); //sent the D to the lcd
                break;
        }

        if(PIR1 & 0x20) //to check the rcif bit as 1
        {
            lcdcmd(0xC8); //send the location of the display
            rxval = RCREG; //rcreg value to rxval variable
            lcddata(rxval); //sent the data to lcd
        }
    }

    void init()
    {
        TRISB=0xF0; //initialize the msb nibble as input
        PORTB=0x00; //clear the port b
        TRISD=0x00; //trisd set as output
        OPTION_REG=0x7F; //enable the internal pull up resistor
        TRISC=0xC0; //set the rx and tx pin as 1
        RCSTA=0x90; //serial port enable and continuous receive mode
        TXSTA=0x20; //transmit enable bit and asynchronous mode baud rate will be set as low speed
        SPBRG=0x09;
        lcdcmd(0x30); //lcd initialization
        __delay_ms(100);
        lcdcmd(0x30); //lcd initialization
        __delay_ms(100);
        lcdcmd(0x30); //lcd initialization
        __delay_ms(100);
    }
}

```

Task program

```

    __delay_ms(100);
    lcdcmd(0x38); //select the twoline and font of the display
    lcdcmd(0x0C); //for display on and cursor off
    lcdcmd(0x01); //display clear
    __delay_ms(100);
}

void lcdcmd(unsigned char j)
{
    PORTB&=~0x01; //rs pin will set as 0
    PORTD=j; //sent the j value to the portd
    PORTB|=0x02; //enable pin set 1
    __delay_ms(50);
    PORTB&=~0x02; //enable pin set 0
}

void lcddata(unsigned char j)
{
    PORTB|=0x01; //rs pin will set as 1
    PORTD=j; //sent the data to the portd
    PORTB|=0x02; //enable pin set as 1
    __delay_ms(50);
    PORTB&=~0x02; //enable pin set as 0
    //__delay_ms(100); //delay
}

```

PROGRAM:-

RECIVER (RX):-

```

#include <xc.h> //incloed the header file
#define _XTAL_FREQ 6000000 //italize the clock speed
void receivee_init(void); //function declaration
void lcdcmd(unsigned char); //function declaration
void lcddata(unsigned char); //function declaration
unsigned char i, recval, rx[10]={"RX DATA="}, tx[10]={"TX DATA="}; //declare the variable
void main()
{
    receivee_init(); //call the recive init function
    lcdcmd(0x80); //set the location of the lcd
    for(i=0; i!=8; i++)
        lcddata(rx[i]); //send the data to lcd
    lcdcmd(0xC0); //set the location of the lcd
    for(i=0; i!=8; i++)
        lcddata(tx[i]); //send the data to lcd
    while(1)
    {
        if(PIR1 & 0x20) //check the rcif bit will 1
        {
            recval = RCREG; //rcreg data will be stored in recval
            switch(recval)
            {
                case 'A':
                    lcdcmd(0x88); //set the location of the lcd
                    lcddata(recval); //send the data to lcd
                    TXREG=recval+32; //to conver the capital to smaller
                    lcdcmd(0xC8); //set the location of the lcd
                    lcddata(recval+32); //send the data to lcd
                    break;
                case 'B':
                    lcdcmd(0x88); //set the location of the lcd
                    lcddata(recval); //send the data to lcd
                    TXREG=recval+32; //to conver the capital to smaller
                    lcdcmd(0xC8); //set the location of the lcd
                    lcddata(recval+32); //send the data to lcd
            }
        }
    }
}

```

Task program

```

break;
case 'C':
    lcdcmd(0x88); //set the location of the lcd
    lcddata(recval); //send the data to lcd
    TXREG=recval+32; //to conver the capital to smaller
    lcdcmd(0xC8); //set the location of the lcd
    lcddata(recval+32); //send the data to lcd
break;
case 'D':
    lcdcmd(0x88); //set the location of the lcd
    lcddata(recval); //send the data to lcd
    TXREG=recval+32; //to conver the capital to smaller
    lcdcmd(0xC8); //set the location of the lcd
    lcddata(recval+32); //send the data to lcd
break;
}
}
}

```

```
void receivee_init()
```

```

{
    TRISD=0x00; //set port d as output
    TRISB=0x00; //set the portb all as output
    TRISC=0xC0; //set the tx and rx pin as 1
    RCSTA=0x90; //serial port enable and continous receive mode
    TXSTA=0x20; //transmitt enable bit and asynchroneus mode boud rate will be set as low speed
    SPBRG=0x09; //set the boud rate
    lcdcmd(0x30); //lcd intialization
    __delay_ms(100);
    lcdcmd(0x30); //lcd intialization
    __delay_ms(100);
    lcdcmd(0x30); //lcd intialization

```

```

    __delay_ms(100);
    lcdcmd(0x38); //select the twoline and font of the display
    lcdcmd(0x0C); //for display on and cursear off
    lcdcmd(0x01); //desplay clear
    __delay_ms(100);
}

```

```
void lcdcmd(unsigned char j)
```

```

{
    PORTB&=~0x08; //rs pin will set as 0
    PORTD=j; //sent the j value to the portd
    PORTB|=0x04; //enable pin set 1
    __delay_ms(50);
    PORTB&=~0x04; //enable pin set 0
}

```

```
void lcddata(unsigned char j)
```

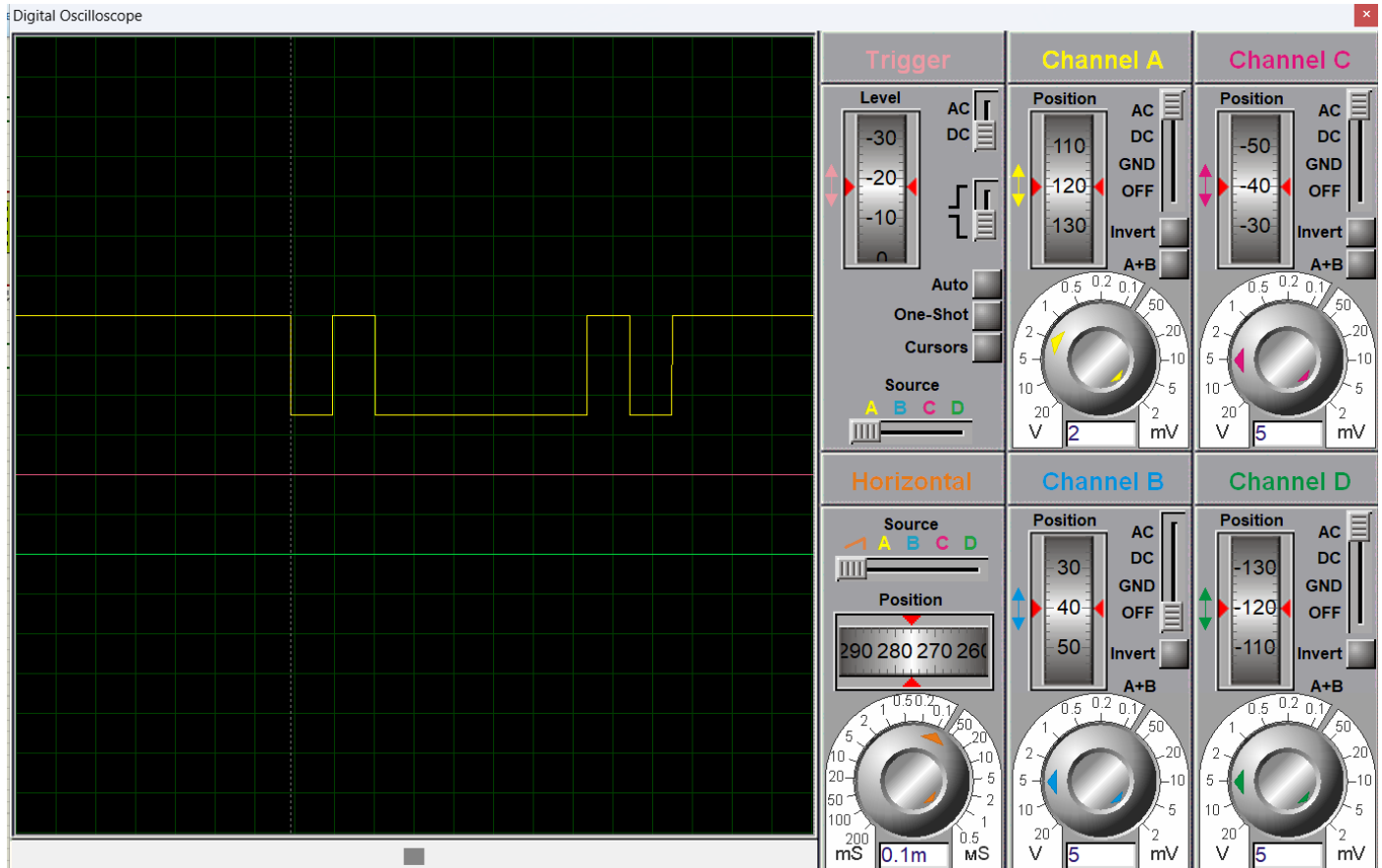
```

{
    PORTB|=0x08; //rs pin will set as 1
    PORTD=j; //sent the data to the portd
    PORTB|=0x04; //enable pin set as 1
    __delay_ms(50);
    PORTB&=~0x04; //enable pin set as 0
    //__delay_ms(100); //delay
}

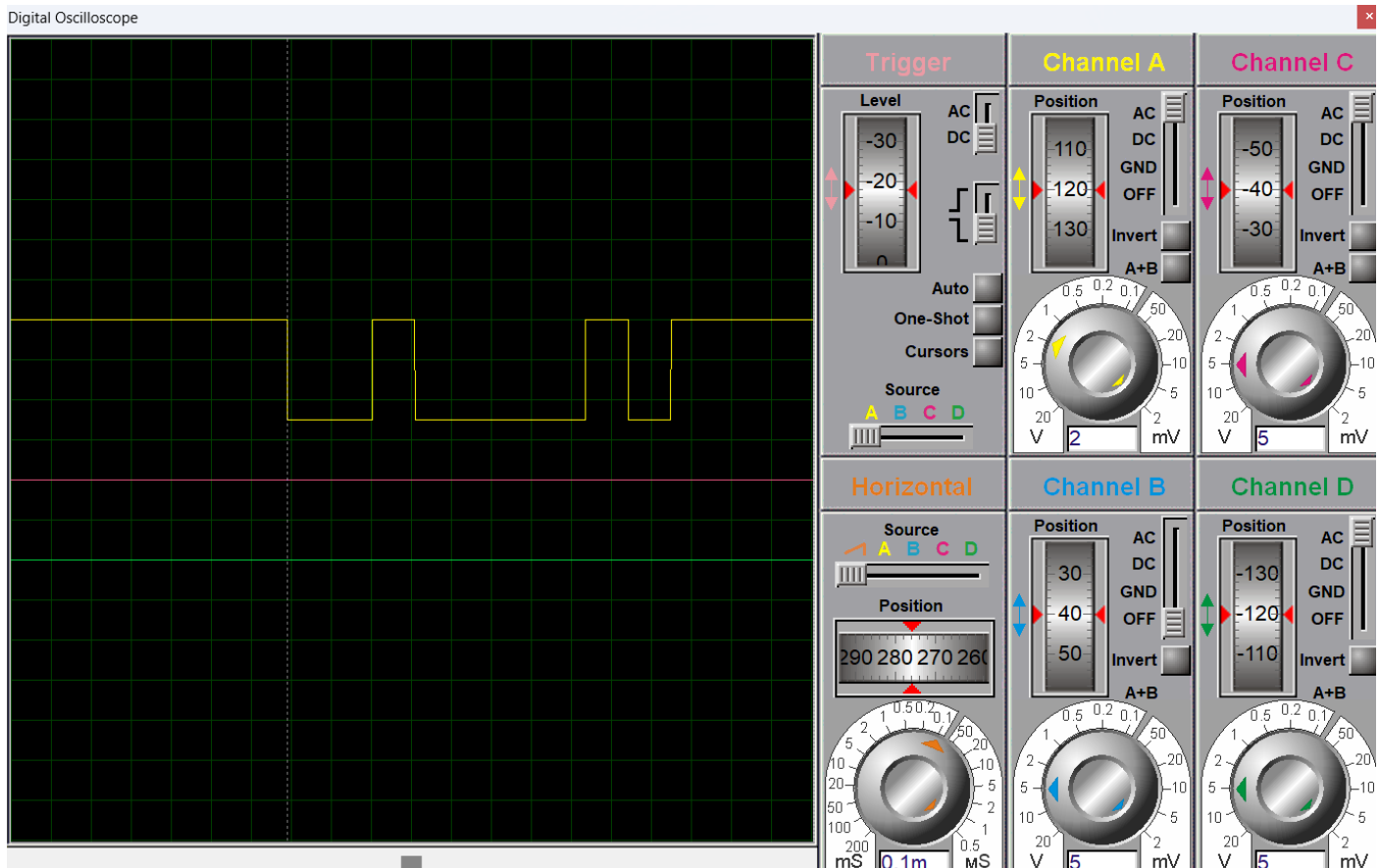
```

UART

Task program OUTPUT WAVE FORM FOR MASTER when transmitting-A



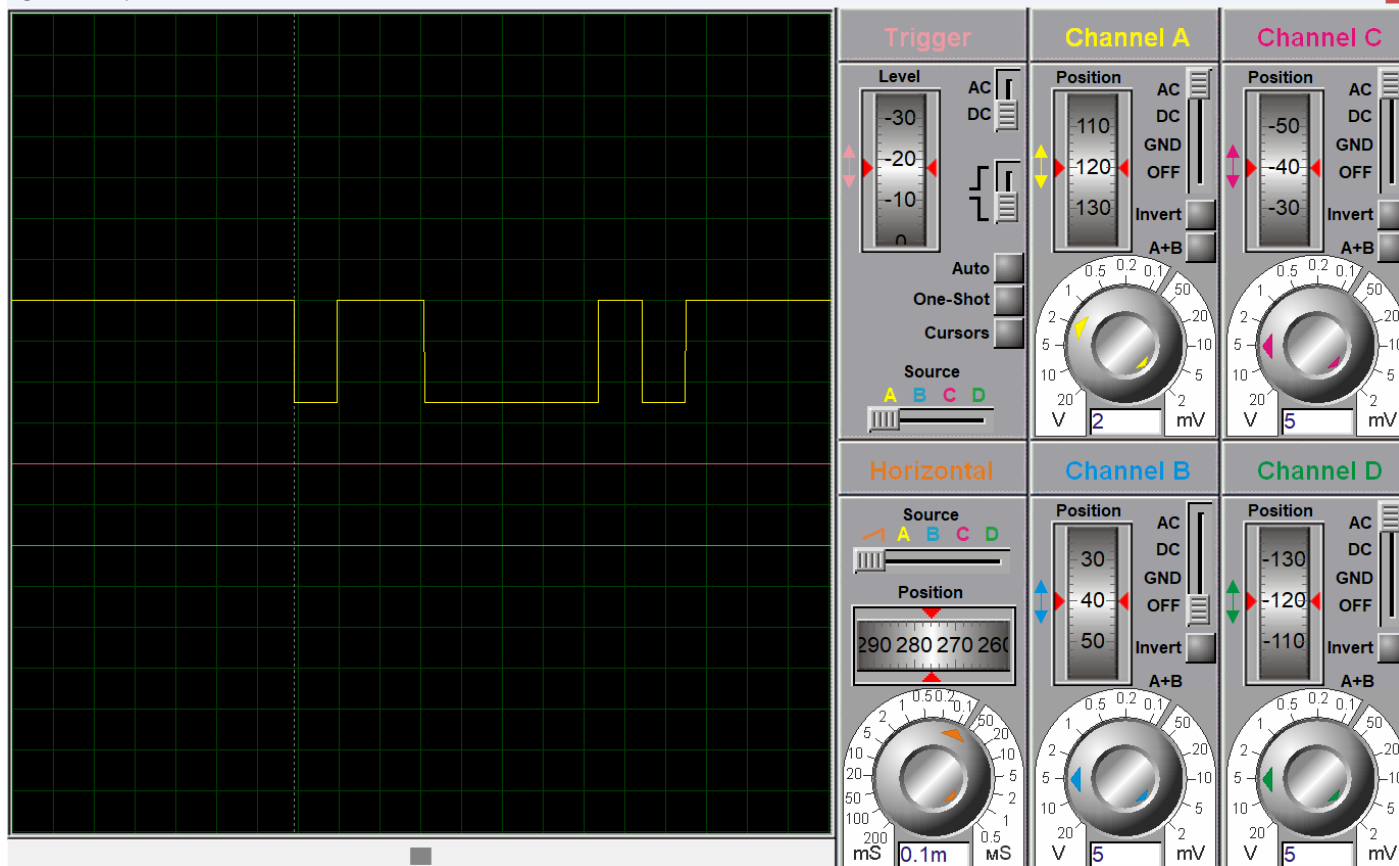
when transmitting-B



UART

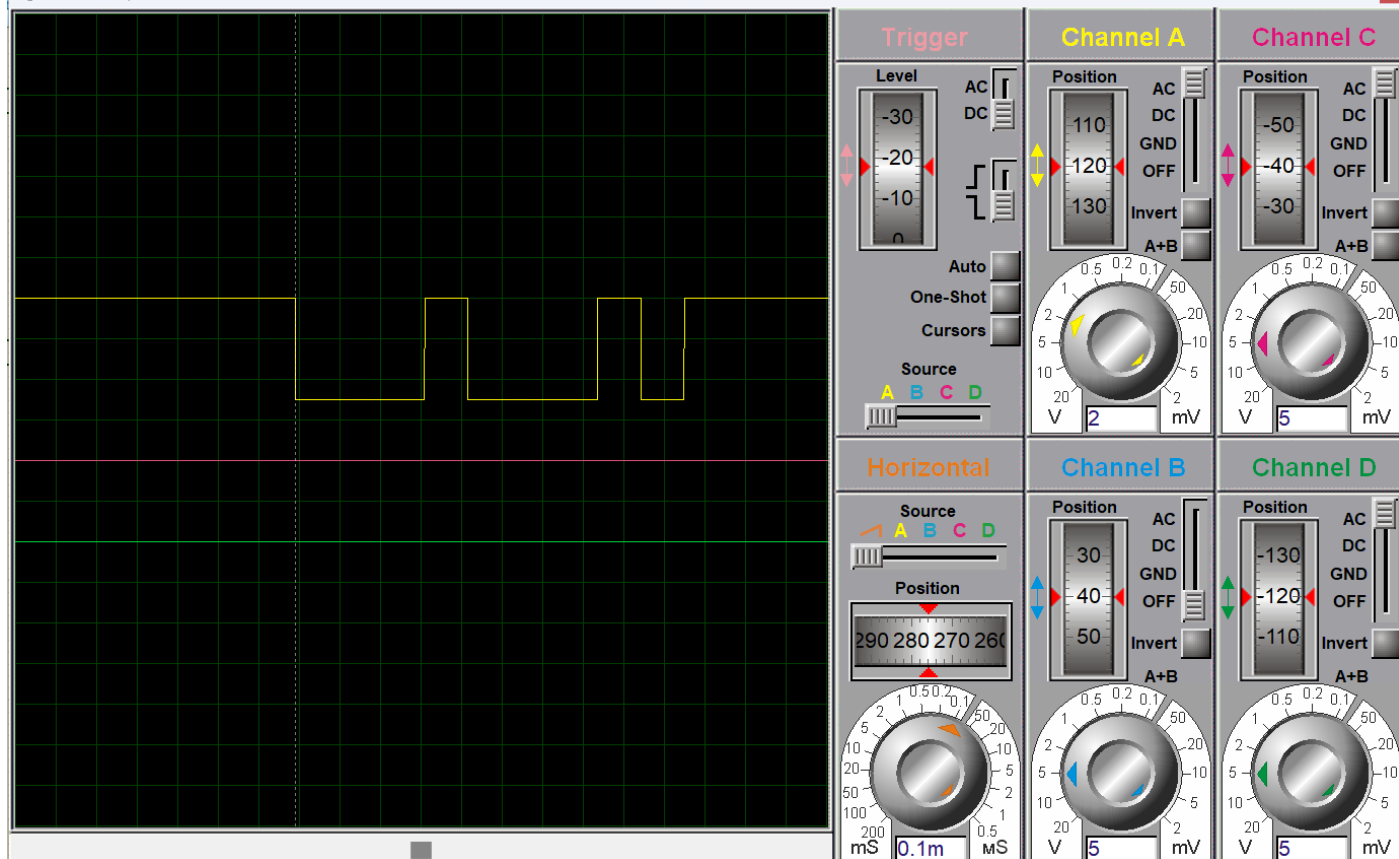
Task program when transmitting-C

Digital Oscilloscope



when transmitting-D

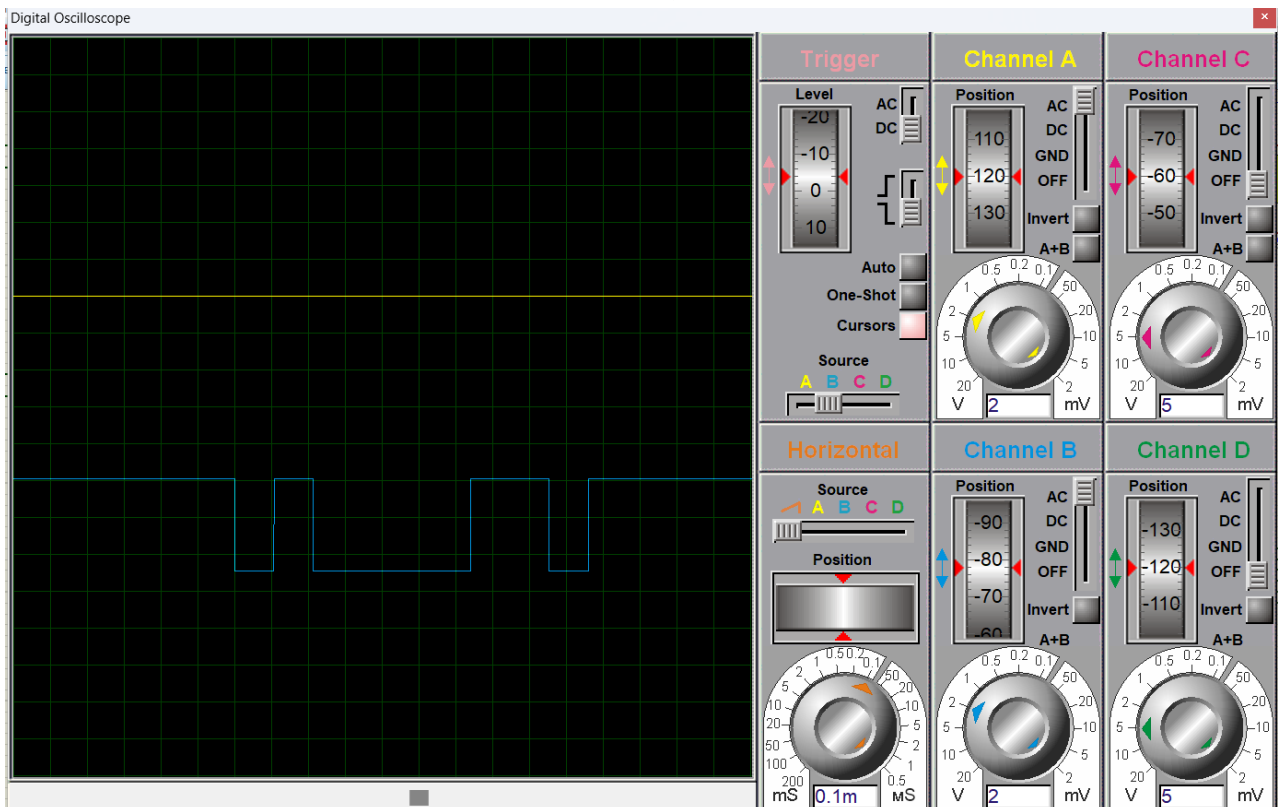
Digital Oscilloscope



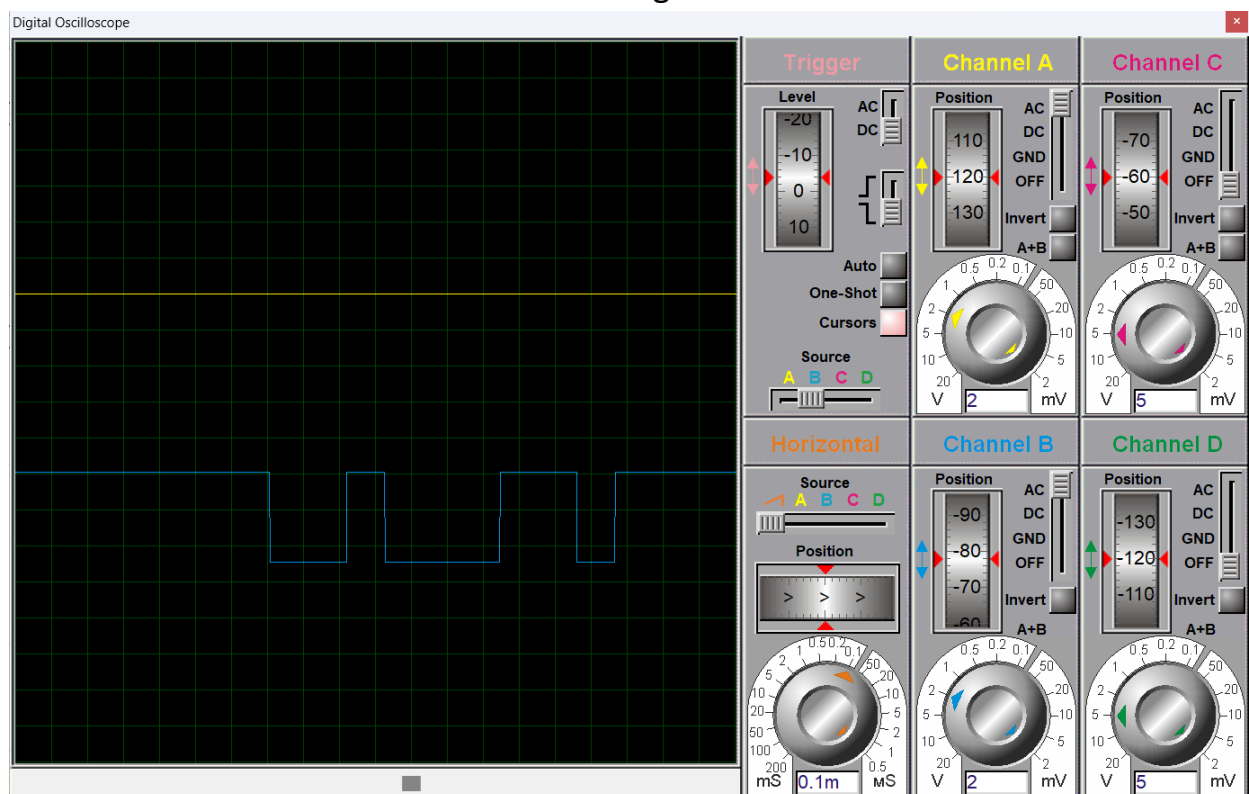
UART

Task program
OUTPUT WAVE FORM FROM SLAVE OR RECIVER OF MASTER

when transmitting **a** from slave

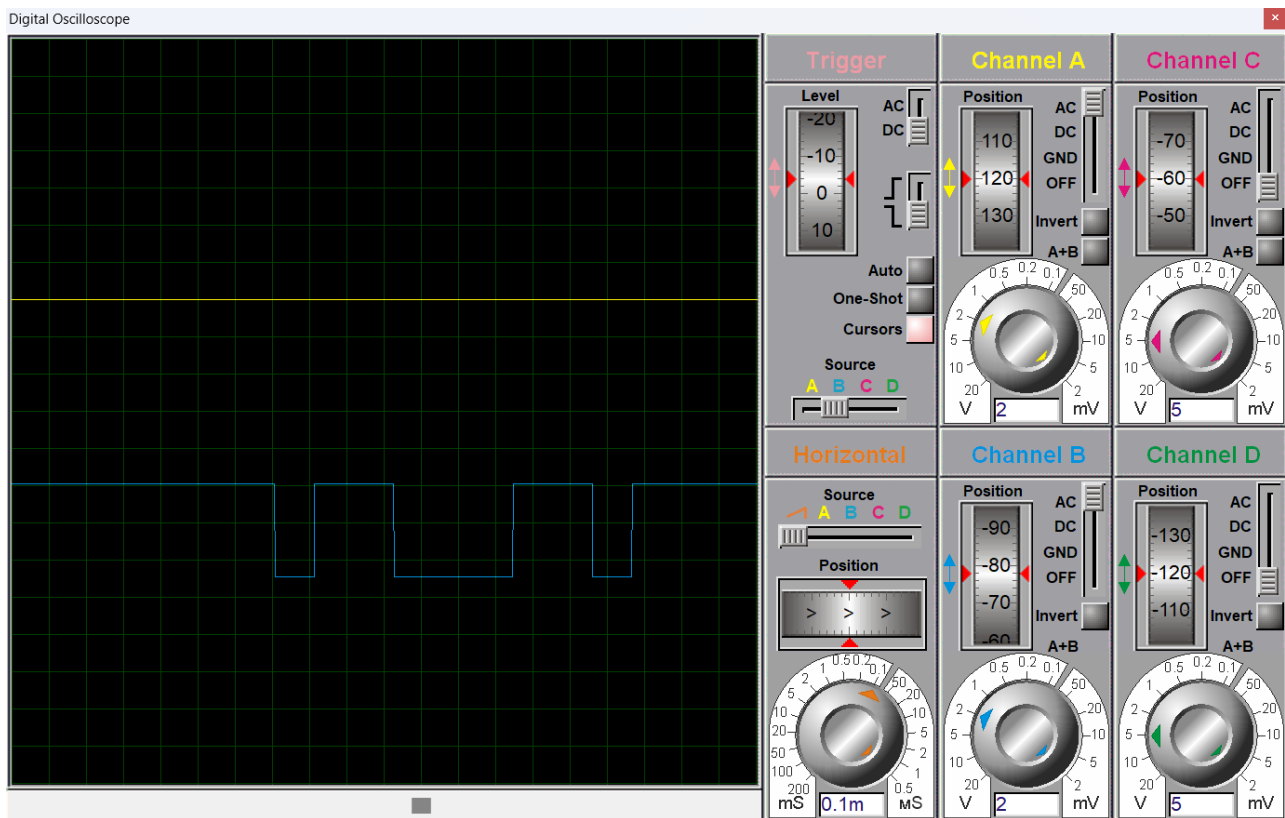


when transmitting **b** from slave



UART

Task program
when transmitting **c** from slave



when transmitting **d** from slave

