

PWM-PULSE WITH MODULATION

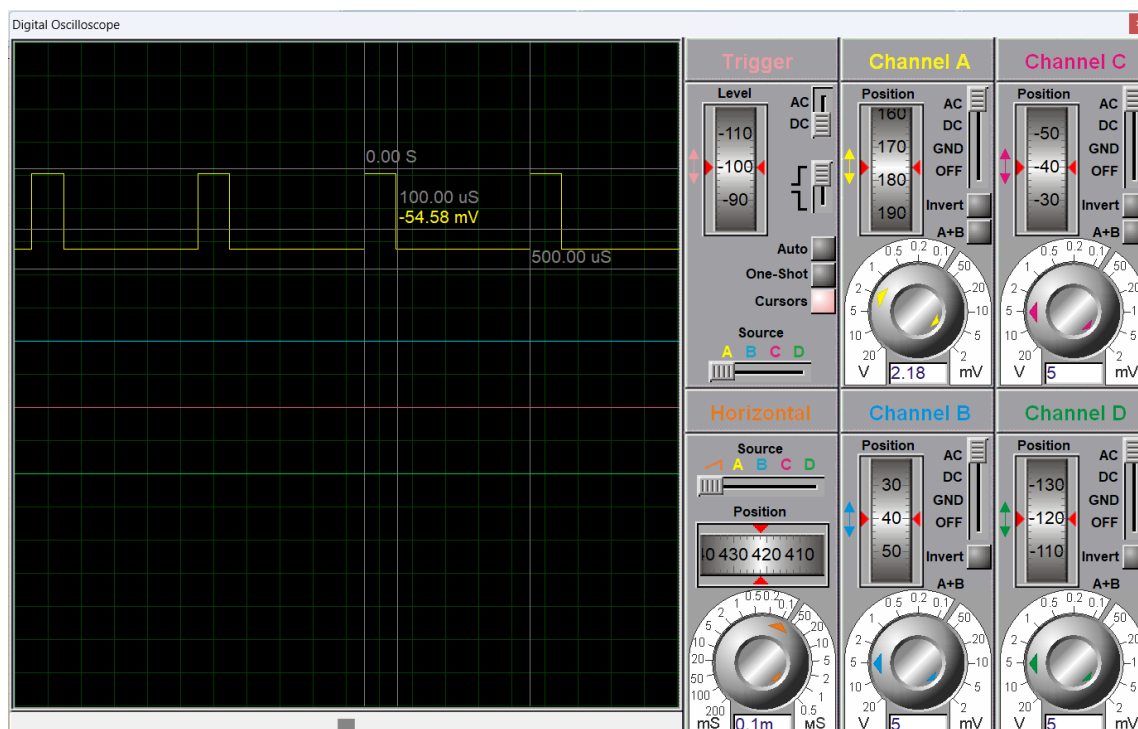
TASK PROGRAM

PROGRAM:-

```
#define _XTAL_FREQ 6000000 //initialize the clock frequency
void init(void); //function declaration
void pwmchange(void); //function declaration
unsigned char lsb20,msb20; //intialize variable
unsigned char lsb60,msb60; //intialize variable
unsigned char lsb90,msb90; //intialize variable
void main() {
    init(); //call init function
    while(1) //while loop
        pwmchange(); //call the pwmchge
}
void init() //init function
{
    TRISC&=~0x04; //portc rc2/ccp1 pen set as output
    CCP1CON=0x0F; //to enter to pwm mode
    T2CON|=0x06; //timer 2 turn on and set pre scaler as 16
    PR2=0x2E; //period will set as 0010 1111
    lsb20=0x20; //store the calculated lsb value to lsb 20
    msb20=0x09; //store the calculate msb value to msb 20
    lsb60=0x10; //store the calculated lsb value to lsb 60
    msb60=0x1C; //store the calculate msb value to msb 60
    lsb90=0x01; //store the calculated lsb value to lsb 90
    msb90=0x2A; //store the calculate msb value to msb 90
}
void pwmchange() //pwmchange function
{
    CCP1RL=msb20; //the msb value will be set to the CCP1L
    CCP1CON=(CCP1CON&~0x30|lsb20); //clear the ccp1con 5th and 4th bit and store lsb20 data
    __delay_ms(3000); //delay
    CCP1RL=msb60; //the msb value will be set to the CCP1L
    CCP1CON=(CCP1CON&~0x30|lsb60); //clear the ccp1con 5th and 4th bit and store lsb60 data
    __delay_ms(3000); //delay
    CCP1RL=msb90; //the msb value will be set to the CCP1L
    CCP1CON=(CCP1CON&~0x30|lsb90); //clear the ccp1con 5th and 4th bit and store lsb90 data
    __delay_ms(3000); //delay
}
```

OUTPUT:-

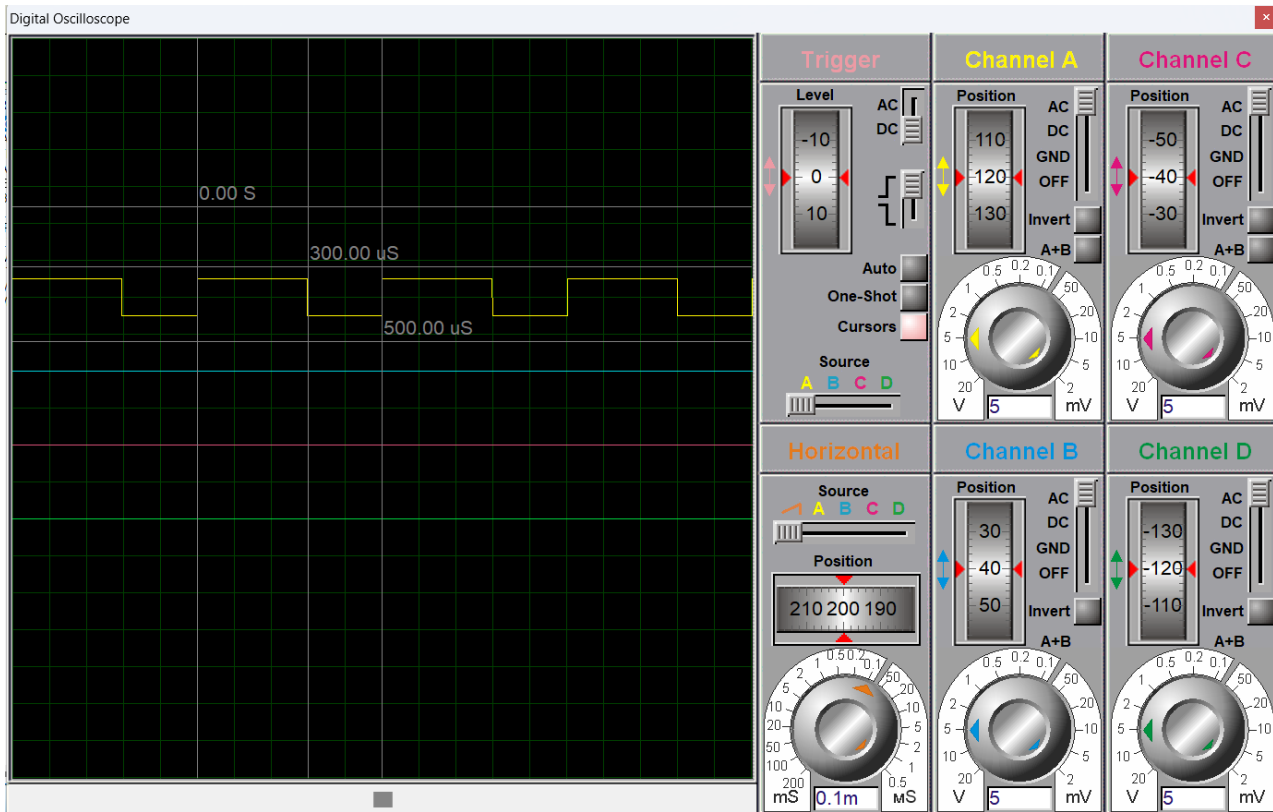
DUTY CYCLE 20%:-



PWM-PULSE WITH MODULATION

TASK PROGRAM

DUTY CYCLE 60%:-



DUTY CYCLE 90%:-

