



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

《单片机原理及应用》作业报告

实验报告 3 第三部分：频率与占空比调节

学院 卓越学院

学号 23040447

姓名 陈文轩

专业 智能硬件与系统(电子信息工程)

2025 年 5 月 6 日

原题目：设计一个简易方波波形发生器，在按键 K1（接 INT0 脚）的触发下可实现 1KHz, 100Hz, 10Hz, 1Hz 的波形切换。用 C 语言编程。（选做：若按键 K2（接 INT1）实现当前频率波形下的占空比可调，如何实现？）

1 实验代码

Code Listing 1: 实验程序

```
1  #include <reg51.h>
2
3  // 宏定义
4  #define FOSC 12000000          // 晶振频率 12MHz
5  #define TIMER_RELOAD_1KHZ (65536 - FOSC / 12 / 2 / 1000 / 2) // 1KHz定时器初
    值 因为是半周期
6  #define TIMER_RELOAD_100HZ (65536 - FOSC / 12 / 2 / 100 / 2) // 100Hz定时器初
    值
7  #define TIMER_RELOAD_10HZ (65536 - FOSC / 12 / 2 / 10 / 2) // 10Hz定时器初
    值
8  // #define TIMER_RELOAD_1HZ (65536 - FOSC / 12 / 2 / 1 / 2) // 1Hz定时器初
    值 250,000超出范围了
9  // 因为是半周期翻转电平 所以再乘以2补偿
10 sbit P2_0 = P2^0; // 方波输出引脚
11 unsigned char mode = 0; // 模式切换变量
12
13 // 外部中断1服务函数
14 void INT0_ISR(void) interrupt 0 {
15     mode = (mode + 1) % 4; // 模式循环切换
16 }
17
18 // 定时器0中断服务函数
19 void Timer0_ISR(void) interrupt 1 {
20     static int counter_1HZ=0; // 基于10HZ的分频，计数十次就是1HZ
21     TH0 = (mode == 0) ? (TIMER_RELOAD_1KHZ >> 8) :
22           (mode == 1) ? (TIMER_RELOAD_100HZ >> 8) :
23           (mode == 2) ? (TIMER_RELOAD_10HZ >> 8) :
24           (TIMER_RELOAD_10HZ >> 8);
25     TL0 = (mode == 0) ? (TIMER_RELOAD_1KHZ & 0xFF) :
26           (mode == 1) ? (TIMER_RELOAD_100HZ & 0xFF) :
27           (mode == 2) ? (TIMER_RELOAD_10HZ & 0xFF) :
28           (TIMER_RELOAD_10HZ & 0xFF);
29 }
```

```
30     if (mode==3) //1HZ
31     {
32         counter_1HZ++;
33         if (counter_1HZ==10)
34         {
35             counter_1HZ=0;
36             P2_0 = ~P2_0;
37
38         }
39     }
40     else
41     {
42         P2_0 = ~P2_0; // 其他模式，直接翻转P2.0引脚电平
43     }
44
45 }
46
47 void main() {
48     // 初始化外部中断1
49     ITO = 1; // 下降沿触发
50     EX0 = 1; // 使能外部中断0
51     EA = 1; // 开启总中断
52
53     // 初始化定时器0
54     TMOD = 0x01; // 定时器0，模式1（16位定时）
55     TH0 = TIMER_RELOAD_1KHZ >> 8;
56     TL0 = TIMER_RELOAD_1KHZ & 0xFF;
57     ET0 = 1; // 使能定时器0中断
58     TR0 = 1; // 启动定时器0
59
60     P2_0 = 0; // 初始化P2.0为低电平
61
62     while (1) {
63         // 主循环，所有逻辑在中断中处理
64     }
65 }
```

2 实验效果

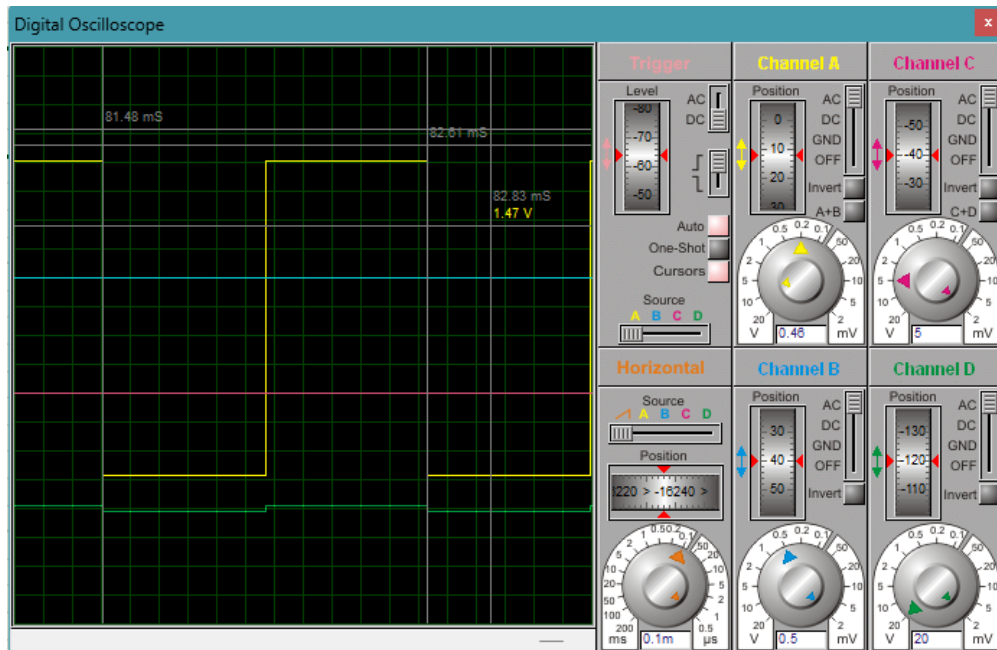


图 1 Proteus 示波器效果，1Khz 方波

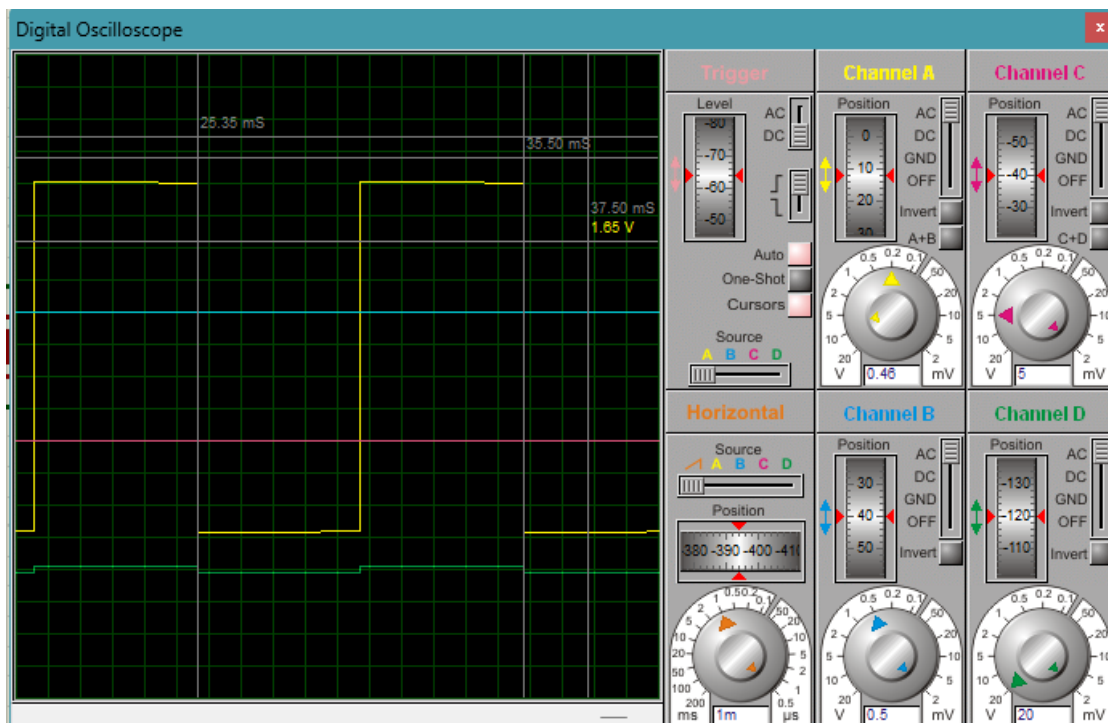


图 2 Proteus 示波器效果，100hz 方波

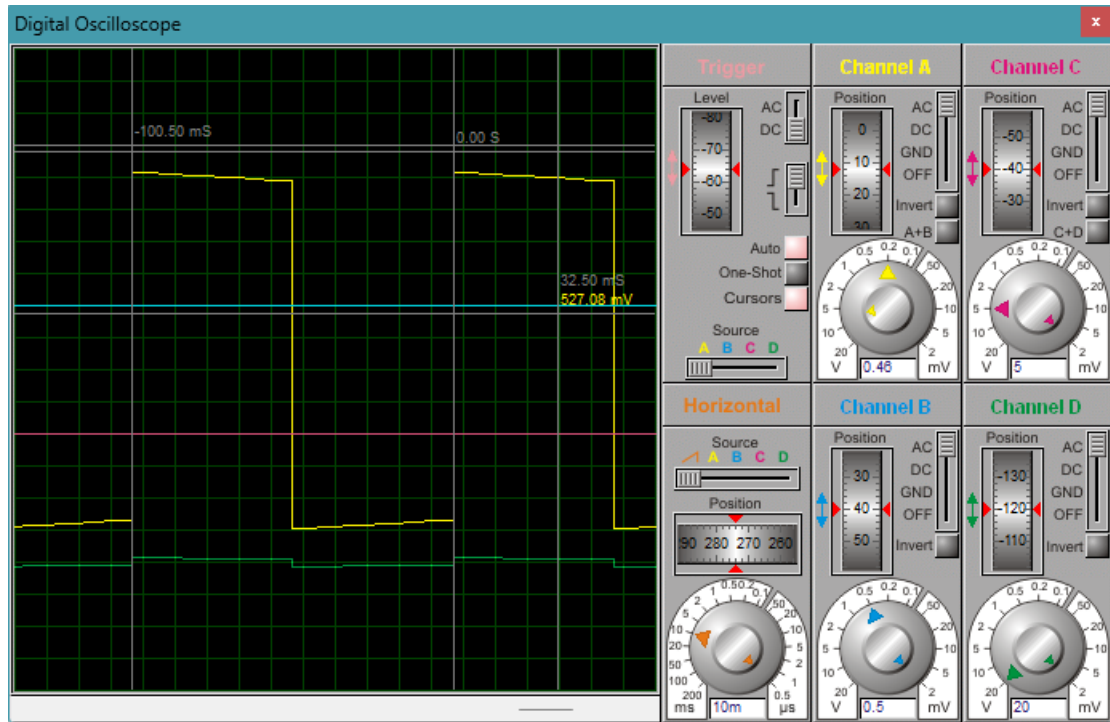


图 3 Proteus 示波器效果，10hz 方波

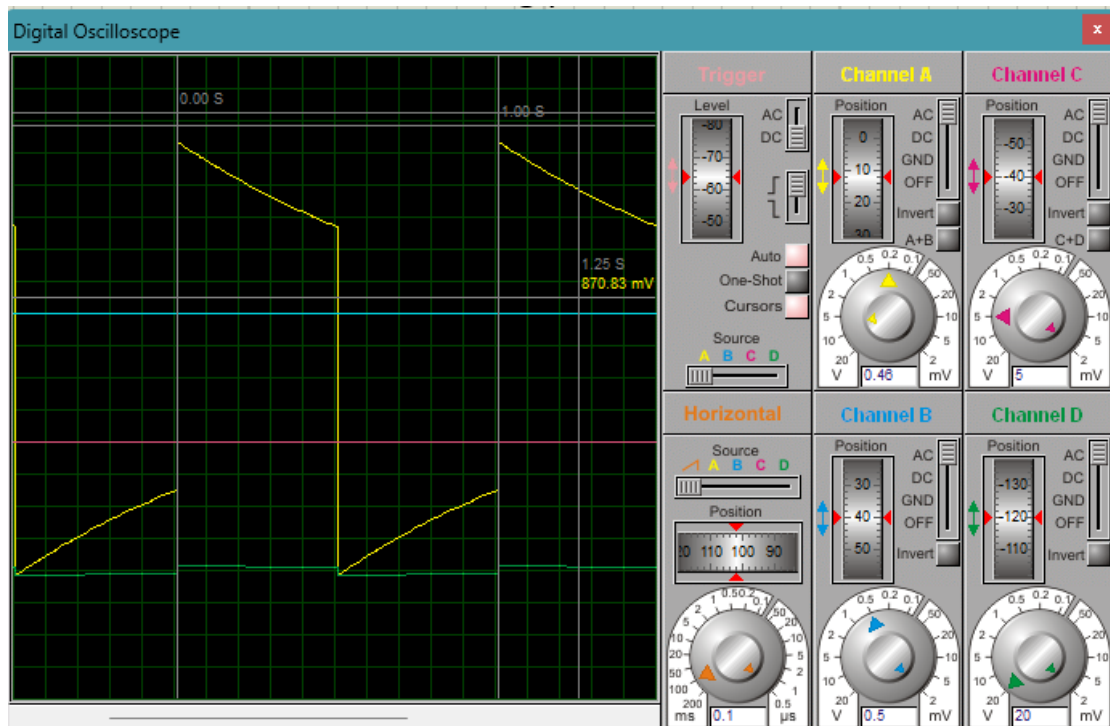


图 4 Proteus 示波器效果，1hz 方波

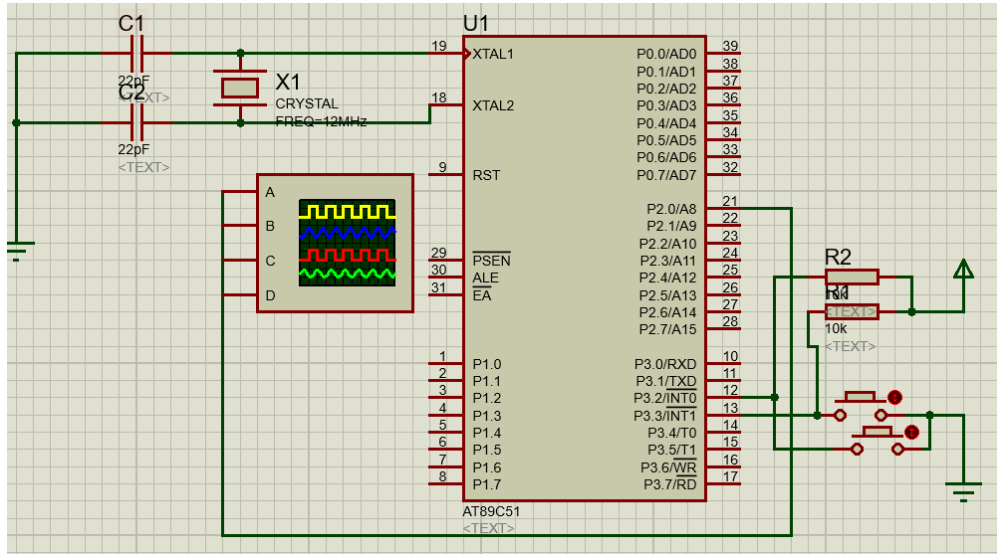


图 5 Proteus 电路结构

3 流程图

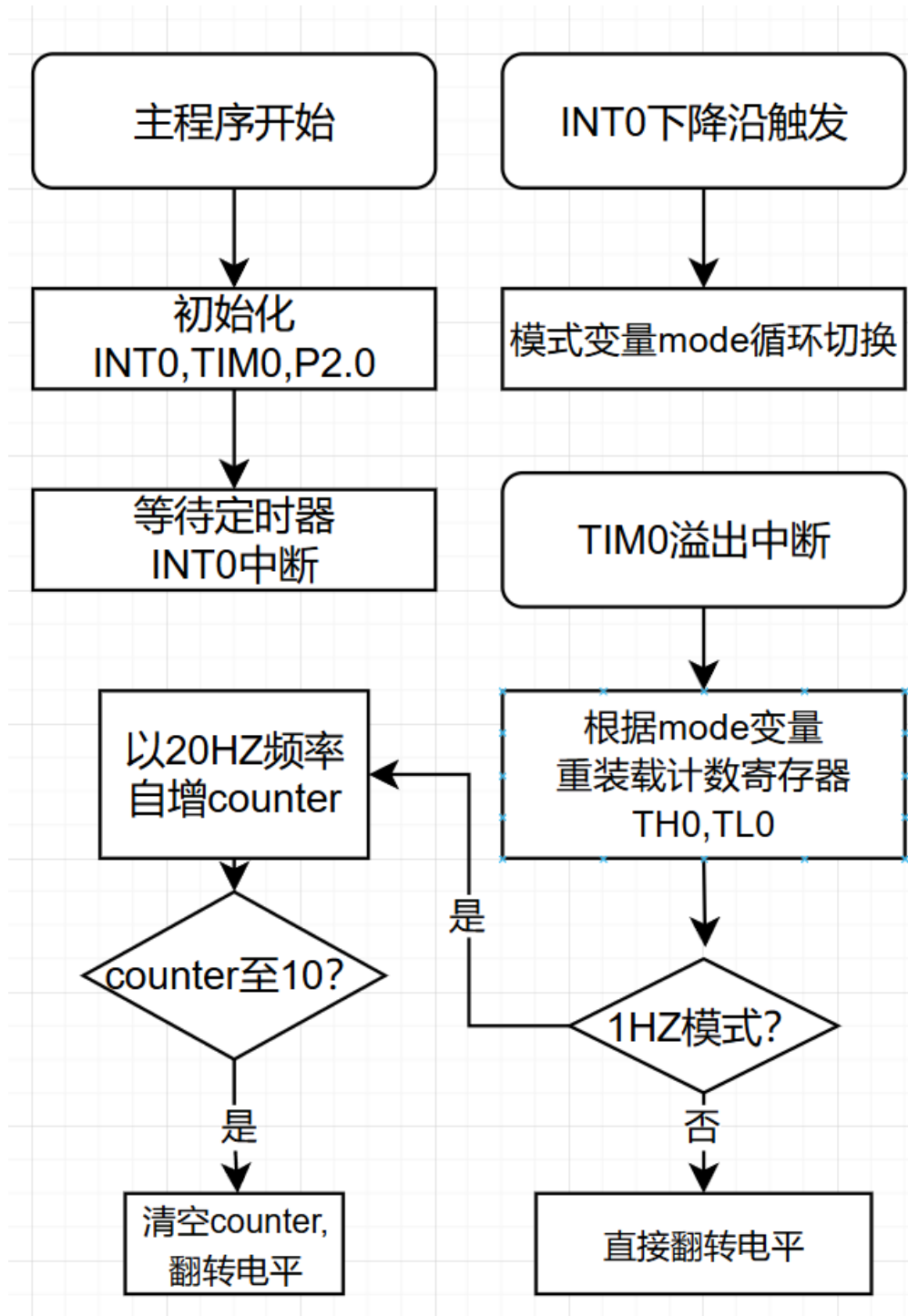


图 6 系统控制流程图

4 选做部分：占空比可调

4.1 实验代码

Code Listing 2: 实验程序

```
1 //拓展程序 占空比可调版本 可以调节10%至90%,步进10%
2 #include <reg51.h>
3
4 // 宏定义
5 #define FOSC 12000000 // 晶振频率 12MHz
6 #define TIMER_RELOAD_1KHZ (65536 - 1) // 1KHz定时器初值 (半周期)
7 //中断程序时间占用 影响频率较大 适当调参 但是周期是1.5ms左右
8 #define TIMER_RELOAD_100HZ (65536 - FOSC / 12 / 2 / 1000) // 100Hz定时器初值
9 #define TIMER_RELOAD_10HZ (65536 - FOSC / 12 / 2 / 100) // 10Hz定时器初值
10 //因为是半周期翻转电平 所以再除以2补偿
11
12 sbit P2_0 = P2^0; // 方波输出引脚
13 unsigned char mode = 0; // 模式切换变量
14 unsigned char duty_cycle = 50; // 占空比, 初始为50%
15
16 // 外部中断0服务函数 (切换频率模式)
17 void INT0_ISR(void) interrupt 0 {
18     mode = (mode + 1) % 4; // 模式循环切换
19 }
20
21 // 外部中断1服务函数 (切换占空比)
22 void INT1_ISR(void) interrupt 2 {
23     duty_cycle += 10; // 占空比增加10%
24     if (duty_cycle > 90) {
25         duty_cycle = 10; // 超过90%后回到10%
26     }
27 }
28
29 // 定时器0中断服务函数
30 void Timer0_ISR(void) interrupt 1 {
31     static unsigned int high_count = 0; // 高电平计数
32     static unsigned int low_count = 0; // 低电平计数
33     static unsigned char state = 0; // 当前状态: 0为高电平, 1为低电平
34     static int counter_1HZ = 0; // 基于10Hz的分频计数
35
36     // 根据模式设置定时器初值
```



```
37 unsigned int timer_reload = (mode == 0) ? TIMER_RELOAD_1KHZ :
38                               (mode == 1) ? TIMER_RELOAD_100HZ :
39                               (mode == 2) ? TIMER_RELOAD_10HZ :
40                               TIMER_RELOAD_10HZ; // 1Hz基
                                                于10Hz分频
41
42 TH0 = timer_reload >> 8;
43 TL0 = timer_reload & 0xFF;
44
45 if (mode == 3) { // 1Hz模式
46     counter_1HZ++;
47     if (counter_1HZ == 10) { // 10Hz分频为1Hz
48         counter_1HZ = 0;
49         if (state == 0) { // 高电平状态
50             high_count++;
51             if (high_count >= duty_cycle / 10) { // 高电平持续时间达到占
                                                空比
52                 high_count = 0;
53                 state = 1; // 切换到低电平
54                 P2_0 = 0;
55             }
56         } else { // 低电平状态
57             low_count++;
58             if (low_count >= (10 - duty_cycle / 10)) { // 低电平持续时间
                                                达到占空比
59                 low_count = 0;
60                 state = 0; // 切换到高电平
61                 P2_0 = 1;
62             }
63         }
64     }
65 } else { // 其他模式
66     if (state == 0) { // 高电平状态
67         high_count++;
68         if (high_count >= duty_cycle / 10) { // 高电平持续时间达到占空比
69             high_count = 0;
70             state = 1; // 切换到低电平
71             P2_0 = 0;
72         }
73     } else { // 低电平状态
74         low_count++;
75         if (low_count >= (10 - duty_cycle / 10)) { // 低电平持续时间达到
                                                占空比
```

```
76         low_count = 0;
77         state = 0; // 切换到高电平
78         P2_0 = 1;
79     }
80 }
81 }
82 }
83
84 void main() {
85     // 初始化外部中断0 (切换频率模式)
86     IT0 = 1; // 下降沿触发
87     EX0 = 1; // 使能外部中断0
88
89     // 初始化外部中断1 (切换占空比)
90     IT1 = 1; // 下降沿触发
91     EX1 = 1; // 使能外部中断1
92
93     // 初始化总中断
94     EA = 1; // 开启总中断
95
96     // 初始化定时器0
97     TMOD = 0x01; // 定时器0, 模式1 (16位定时)
98     TH0 = TIMER_RELOAD_1KHZ >> 8;
99     TLO = TIMER_RELOAD_1KHZ & 0xFF;
100    ETO = 1; // 使能定时器0中断
101    TRO = 1; // 启动定时器0
102
103    P2_0 = 0; // 初始化P2.0为低电平
104
105    while (1) {
106        // 主循环, 所有逻辑在中断中处理
107    }
108 }
```

此代码存在问题：1KHZ 时频率不准，中断程序里面东西太多了，消耗了大量机器周期；1KHZ 时无法调节占空比，其余都可以；其余频率都有一些误差，100HZ 会略微偏小，还是中断程序里面东西太多的原因。

4.2 实验效果

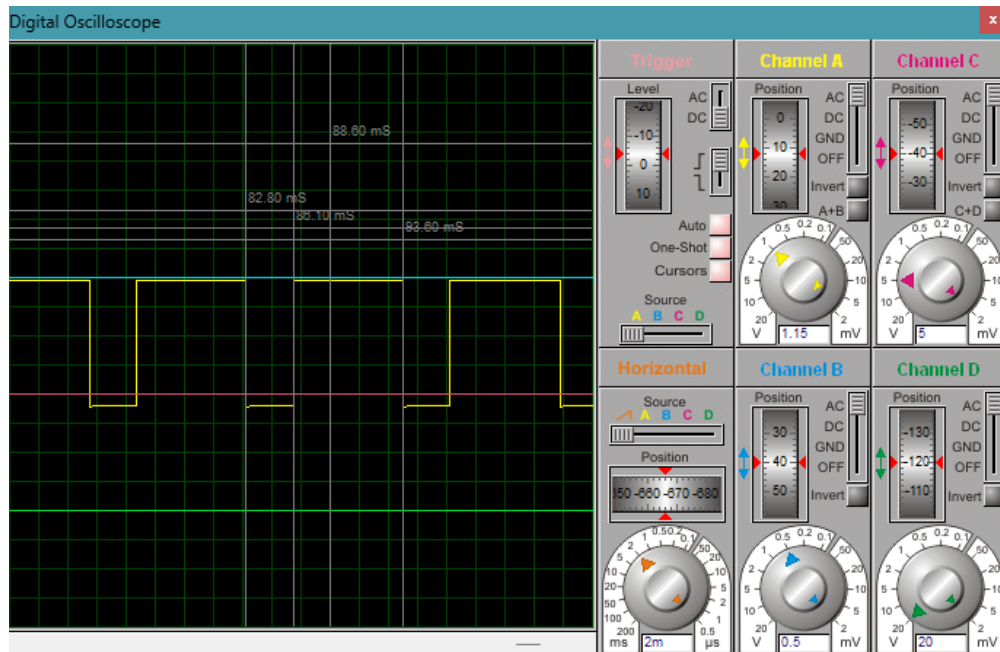


图 7 100HZ,70% 占空比

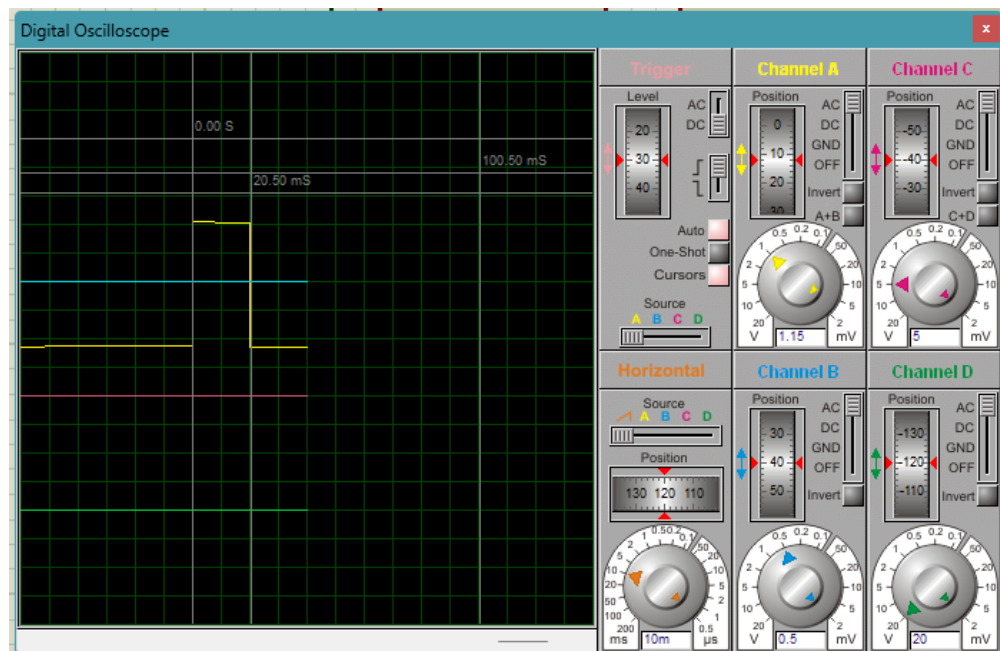


图 8 10HZ,20% 占空比

5 实验体会

本次实验通过单片机定时器和中断的结合，实现了多频率方波的生成及占空比调节功能。实验过程中，发现 1kHz 频率下因中断程序复杂导致频率不准，优化中断代码是关键。此外，占空比调节功能在低频下表现较好，但高频时误差较大，需进一步优化定时器初值计算和中断处理效率。通过实验，初步理解了单片机定时器、中断机制及其对实时性的影响，提升了硬件编程能力。