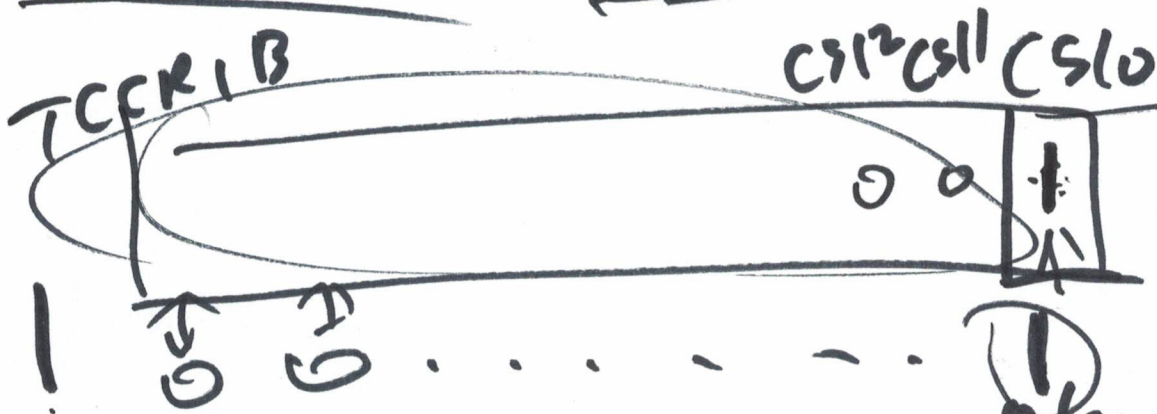


CS12 ~~CS11~~ CS10 TCCR1B

1 < CS10 0 0 0 0 0 0 0 1

CS11

TCCR1B = (1 < CS10) 0 0 0 0 0 0 0 1



TCCR1B = 0x00;

PIC Microchip

TCCR1B.CS10 = 1

STM32

①

while (— == High);

while (— == Low);

while (— == High)

{ TCCR1B |= (1 << CS12);

TCNT1 = 0x10000 - [];

while ((TIFR1 & (1 << TOV1)) == 0)

{ TCCR3B |= (1 << CS30);

TCNT1 = 0x10000

- [];

while ((TIFR3 & (1 << TOV3)) == 0) {

TCCR3B = 0x00;

TIFR3 |= (1 << TOV3);

udisiteWrite (— , ! —);

TCCR1B = 0x00;

TCCR1B |= (1 << TOV1);

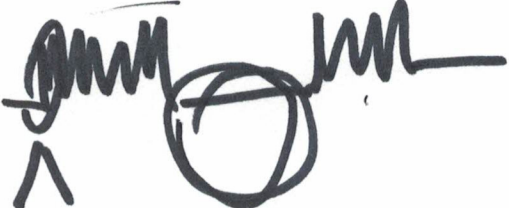
④

μs Delay (—);

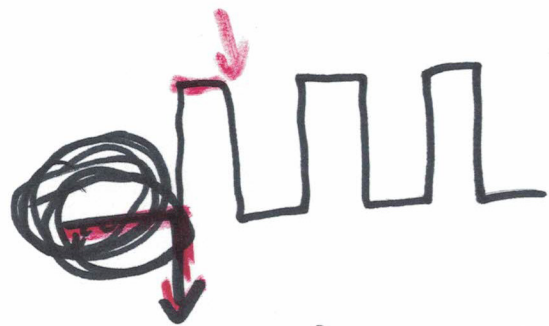
(2)

$$0.55 \times \frac{16M}{256} = 1$$

1:256 prescaler
0.55



μs Delay



while (CTIFR1 & (1 < TOV1)) = 20);

TOV1

CTIFR1



```

#define T_BUTTON_PIN 8
#define B_BUTTON_PIN 9
#define SPEAKER_PIN 5

#define N 7
uint16_t halfPeriodTimer[N] = {10208, 7648, 7648, 7648, 6816, 6080, 6080}; // 1:1 timer counts
uint8_t i = 0;

void setup() {
    pinMode(T_BUTTON_PIN, INPUT);
    pinMode(B_BUTTON_PIN, INPUT);
    pinMode(SPEAKER_PIN, OUTPUT);

    // Timer1 setup (16-bit)
    TCCR1A = 0; // Normal mode
    TCCR1B = 0; // Stop timer
    TCNT1 = 0x10000 - halfPeriodTimer[i]; // Load the counts
}

void loop() {
    //Top button, uses the microSecondDelay() function.
    if (digitalRead(T_BUTTON_PIN) == LOW) {
        while (digitalRead(T_BUTTON_PIN) == LOW) {
            microSecondDelay(halfPeriodTimer[i] / 16); //No. of counts divided by counts/second to find
            time to finish the counts.
            digitalWrite(SPEAKER_PIN, !digitalRead(SPEAKER_PIN)); //Toggle the speaker pin.
        }
        i = (i + 1) % N; //Cycle back to the first element when it reaches the end.
    }

    //Bottom button, uses Timer 1
    if (digitalRead(B_BUTTON_PIN) == LOW) {
        while (digitalRead(B_BUTTON_PIN) == LOW) {
            TCCR1B |= (1 << CS10); // CS10 means timer 1 and the 0 bit. Prescaler set to 1 and also start
            the timer. Or use TCCR1B |= B00000001
            while ((TIFR1 & (1 << TOV1)) == 0); //While counting, TOV1 is 0
            TCCR1B = 0x00; //Stop the timer when it gets out of the While loop
            TIFR1 |= (1 << TOV1); //Write a 1 to TOV1 to reset TOV1 to 0 (I know this sounds weird).
            TCNT1 = 0x10000 - halfPeriodTimer[i]; //Reload the counts.
            digitalWrite(SPEAKER_PIN, !digitalRead(SPEAKER_PIN));
        }
        i = (i + 1) % N;
    }
}

void milliSecondDelay(uint16_t ms) {
    for (uint16_t i = 0; i < ms; i++) {
        microSecondDelay(1000);
    }
}

void microSecondDelay(uint16_t us) {
    for (uint16_t i = 0; i < us; i++) {
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
    }
}

```

counts to be finished