

Using the HT45R36 I/O for Buzzer Driving

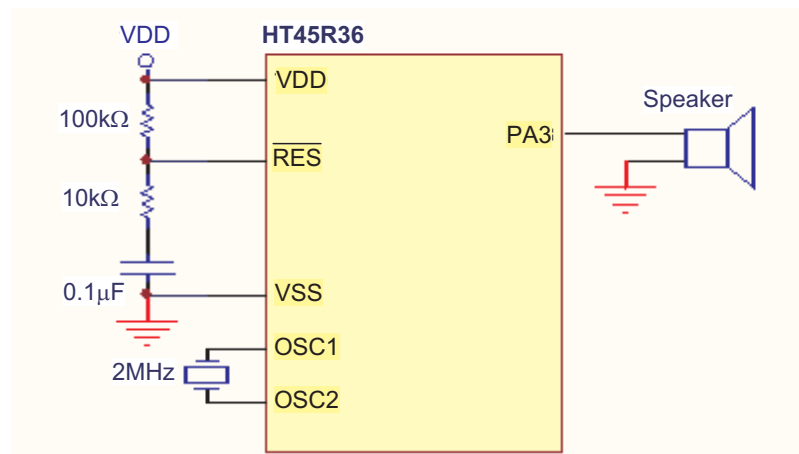
D/N: HA0119E

Introduction

Some MCU devices do not possess a PFD function, however their applications may require a buzzer function. Therefore this application will use the HT45R36 as an example to show how a normal I/O pin can be used to simulate a PFD function and to drive a buzzer at a frequency of 2kHz.

Application Circuit

Connect a buzzer directly between the I/O pin PA3 and ground. By setting up the pin as an input or output the on/off function of the buzzer can be controlled.



Software Description

```

.listinclude
INCLUDE HT45R36.INC
;*****
DATA_MAIN .SECTION 'DATA'      ;define data registers
R_ACC_BAK      DB ?      ;accumulator backup
R_STATUS_BAK   DB ?      ;status backup
R_COUNTER1     DB ?      ;Counter 1
R_COUNTER2     DB ?      ;Counter 2
F_HALFSECOND   DBIT      ;0.5s status register
;-----
---
CODE_MAIN .SECTION AT 00H 'CODE';Program Section
    JMP MLP_MAIN
    ORG 04H      ;Interrupt 0 entry point
    RETI
    ORG 08H      ;Interrupt 1 entry point
    RETI
    ORG 0CH      ;Timer/Event counter entry point
    JMP ISR_TIMER ;jump to timer routine
    ORG 10H      ;RC oscillator entry point
    RETI

ISR_TIMER:      ;Timer routine
    MOV R_ACC_BAK,A      ;backup accumulator
    MOV A,STATUS
    MOV R_STATUS_BAK,A   ;backup status register
    MOV A,08H
    XORM A,PA      ;PA3 output inverted
    INC R_COUNTER1      ;increase interrupt counts by 1
    MOV A,R_COUNTER1
    XOR A,80
    SNZ Z      ;is time up (80*125us)=10ms?
    JMP L_EXIT_ISR_TIMER ;not yet at 10ms, exit interrupt
    CLR R_COUNTER1      ;time up, clear timer
    INC R_COUNTER2      ;if up to 10ms - increment this register
    MOV A,R_COUNTER2
    XOR A,50
    SNZ Z      ;is time up (50*10ms)=0.5s?
    JMP L_EXIT_ISR_TIMER ;not yet at 0.5s, exit interrupt
    CLR R_COUNTER1      ;time up, clear timer
    CLR R_COUNTER2      ;time up, clear timer
    SZ F_HALFSECOND     ;check 0.5s flag
    JMP L_CLRFLAG       ;0.5s flag is set, jump to clear 0.5S flag
    SET F_HALFSECOND     ;set 0.5s flag
    JMP L_EXIT_ISR_TIMER ;exit interrupt
L_CLRFLAG:
    CLR F_HALFSECOND     ;clear 0.5s flag
    
```

```

L_EXIT_ISR_TIMER:
    MOV    A,R_STATUS_BAK    ;restore status register
    MOV    STATUS,A
    MOV    A,R_ACC_BAK      ;restore accumulator
    RETI
MLP_MAIN:
    MOV    A,09H            ;
    MOV    INTC0,A          ;enable timer interrupt
    CLR    F_HALFSECOND    ;initialise 0.5s flag
    CLR    R_COUNTER1      ;initialise timer register 1
    CLR    R_COUNTER2      ;set PA3 as an output, close PFD
    MOV    A,06H
    MOV    TMR,A            ;set timer initial value to 6, count
                                ; to 250 and generate interrupt

    MOV    A,91H
    MOV    TMRC,A          ;setup internal timer,
                                ;division ratio MHz/2=1MHz
                                ;every 250us generate an interrupt

LOOP:
    SZ     F_HALFSECOND    ;check 0.5s flag set
    JMP    L_PFDOPEN       ;if set jump to start PFD
L_PFDCLOSE:
    SET    PAC.3           ;if not set set PA3 as input and stop PFD
    JMP    LOOP            ;cycle check output is PFD
L_PFDOPEN:
    CLR    PAC.3           ;if yes, set PA3 as output and start PFD
    JMP    LOOP            ;cycle check if output is PFD

```

Software Description

This program function will emit a sound every 0.5s for a time period of 0.5s.

The program uses an interrupt period of 250us and uses the PA3 output. PA3 will emit a frequency of 2kHz and drive a buzzer.

Until PA3 is setup as an output, PA3 has not way to drive the buzzer and therefore the buzzer will stop emitting sounds.

The program uses the Timer/Event Counter R_COUNTER1 , R_COUNTER2 to setup a time of 0.5s. When a 0.5s time has elapsed, the 0.5s time flag will invert, which will in turn cycle the on/off function of the buzzer.