

AT3223 Station Master

Atmel AVR 2006 Design Contest

Project Number: AT3223

Project Title: Station Master

AVR Component: ATtiny13

Abstract

Station Master

Overview

The Station Master is a small key ring sized alarm that uses an accelerometer to track your movement. The device will gently vibrate and let you know when you've reached your stop and it's time to disembark from the train. Using a simple one button interface the user can set an alarm for the right number of stops and then be reminded when its time to get off.

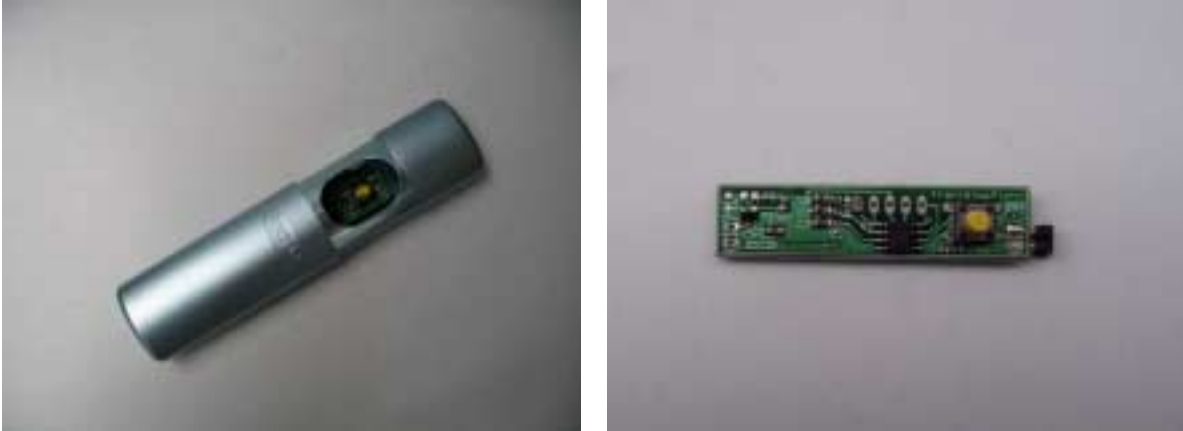
The ATtiny13 was selected for this design due to its low cost and suitable array of features. An MMA7260 3-Axis Accelerometer from Freescale is used to monitor the tilt and movement of the device. A simple cell phone's alarm motor is used to provide a discrete silent alarm. These motors can be found very cheap at electronic surplus stores.

Due to the high computational needs of integrating acceleration data and calculating actual speed and distance in real-time this initial approach was quickly discarded in favour of a simpler design. After analysing data from many train journeys (using a data-logger and accelerometer) I was able to develop a simple algorithm to establish whether the train was moving or stationary. It was important that this algorithm could work with only 64 Bytes of RAM available on the ATtiny13; the final implementation used only 6 Bytes of RAM. By programming in assembler I was able to pack plenty of code into the flash and still have room for a 256-byte sine wave table. The sine wave is used to give the alarm a really natural look and feel when using either an LED or motor for output.

Although this design worked for my own daily commute its main filter would likely need to be re-tuned for other train systems. The best application for this device would be to incorporate it into a cell phone or portable music device (especially if they already uses an accelerometer for HDD protection) that way existing system resources could be used to implement a learning function, or more accurate stop detection as well as reducing the number of gadgets you need to carry. Imagine an MP3 player that fades out the music and announces your stop at the appropriate time!

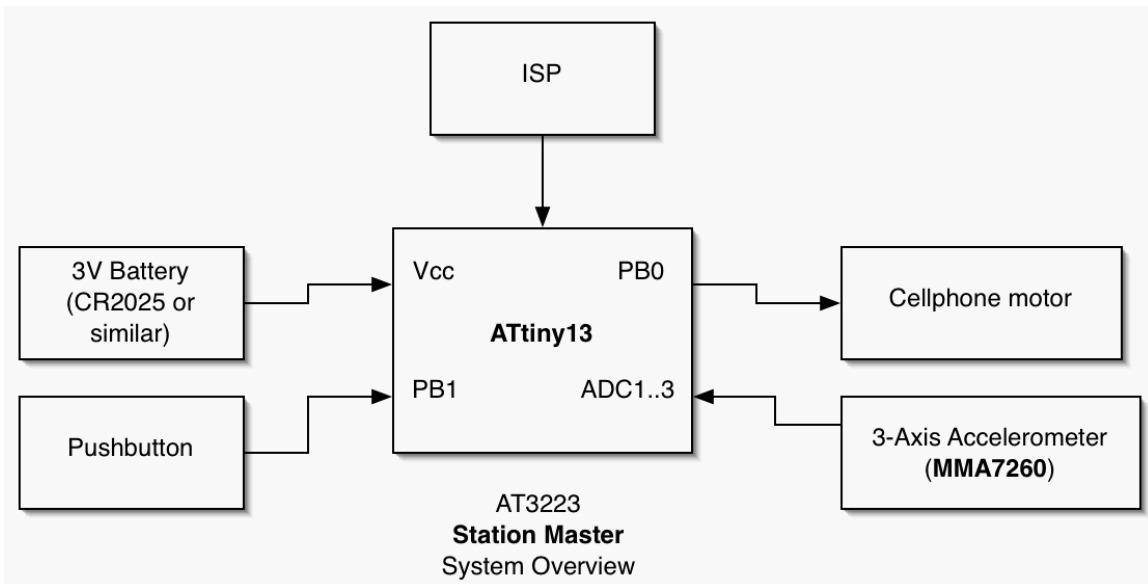
AT3223 Station Master

Photos



The left photo shows the prototype in its discrete case that slides open to reveal the button, and slides shut again to prevent accidental triggering. The right photo displays the tiny PCB. The accelerometer is on the reverse side.

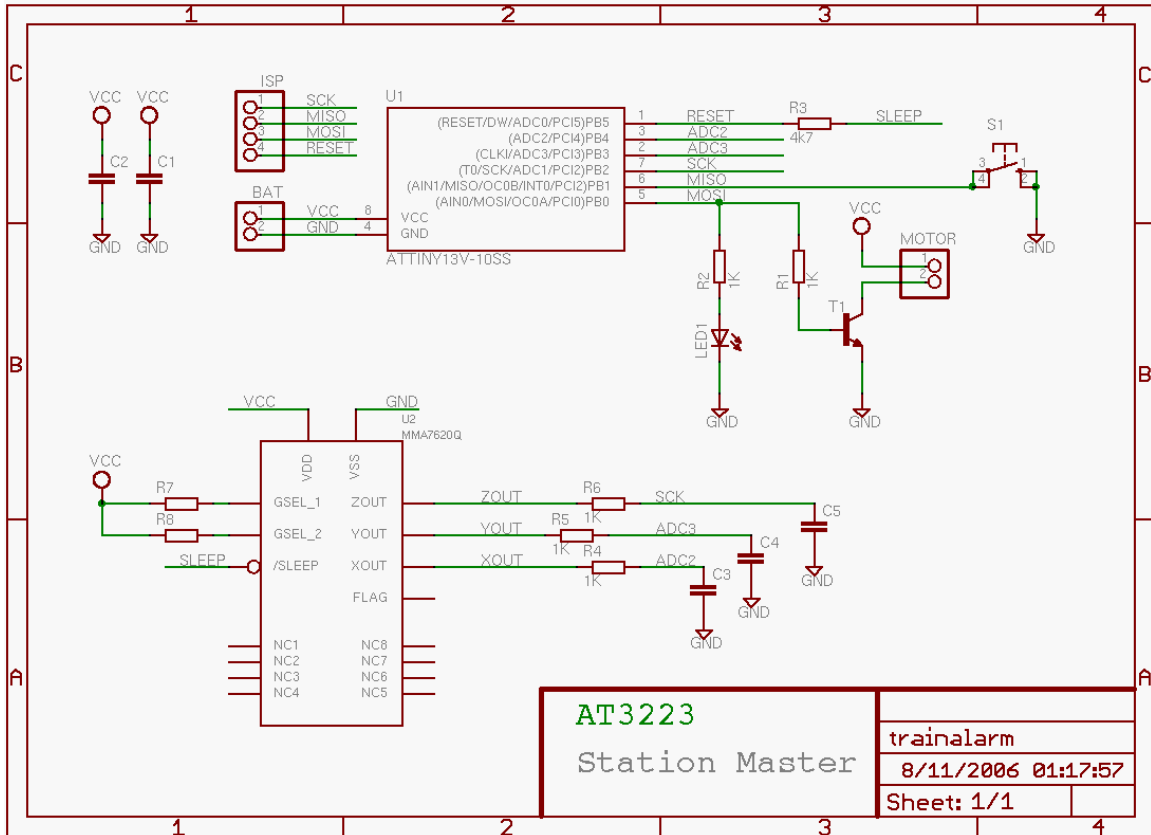
Block Diagram



The overall system operation is very simple. Data is read from the pushbutton and the accelerometer and then once certain conditions are met the cell phone motor is gently pulsed with a sine wave. Clicking the button wakes the device and adds a stop to the alarm counter while holding the button for one second disables the alarm and puts the device to sleep.

AT3223 Station Master

Schematic



Although the circuit is very simple, the ATtiny13 is almost at its limit in terms of available IO pins. In order for ISP programming to work reliably C5 must be disconnected during programming. The circuit also has provision for an LED based visual alarm as an alternative to the vibrating cell phone motor. In a finished device the reset pin would be used as an I/O pin to put the accelerometer into sleep mode when not in use.

AT3223 Station Master

Code Sample

```
*****  
; MAIN LOOP  
;  
loop:  
;check for alarm status  
check_alm:  
    sbrs state,sALARM  
    rjmp update_motor  
  
;Update motor for alarm state  
alarm_motor:  
    out OCR0A,pwm ;set motor power  
    rjmp check_button1  
  
;Update motor for normal state  
update_motor:  
    sbrs state,sMOTOR  
    rjmp motor_OFF  
    out OCR0A,pwm ;set motor power  
    rjmp check_button1  
motor_OFF:  
    clr tempb ;turn motor off  
    out OCR0A,tempb  
;    out OCR0A,f_lnoiseCnt ;FOR DEBUG>> REMOVE ME LATER  
;    out OCR0A,f_sampleCnt ;FOR DEBUG>> REMOVE ME LATER  
  
;Read buttons  
check_button1: ; control switch  
    sbrs state,sBUTTON1  
    rjmp update_filter  
  
    ; Button 1 pushed -> add station to count.  
    ;clear button press flag ;clear alarm if it is sounding  
    cbr state,(1<<sBUTTON1)|(1<<sIDLE)|(1<<sALARM)  
  
;Add another station to the countdown.  
add_station:  
    ;reset timers  
    clr countL  
    clr countH  
    sbr state,(1<<sLONG) ;Flag for long key press  
  
    ;increment the alarm setting  
    ldi tempa,STATION_INC  
    add alarm,tempa  
    sbr state,(1<<sMOTOR) ;pulse to acknowledge  
    ldi ZL,PWM_ZERO ;soft fade in  
  
;Check if motor is off and read adc if an update is due  
update_filter:  
    ldi tempa,(1<<sMOTOR)|(1<<sALARM)  
    and tempa,state ;dont read acccel when motor is on  
    brne end  
    sbrs state,sREADADC ;check if an update is due  
    rjmp end  
    cpse alarm,zeroReg ;dont update if no alarm is set  
  
    rcall filter ;read ADC and update f_sate etc.  
    cbr state,(1<<sREADADC) ;clear the flag
```

(Continued on next page)

AT3223 Station Master

(Continued from previous page)

```
end:
; Go to sleep and wait for an interupt to wake us.
; (IDLE MODE) keep timers running. stop main clock.
sei
ldi  tempa,(0<<PUD)|(1<<SE)|(0<<SM1)|(0<<SM0)|(0<<ISC01)|(0<<ISC00)
out  MCUCR,tempa
sleep

; just woke up, disable sleep
ldi  tempa,(0<<PUD)|(0<<SE)|(0<<SM1)|(0<<SM0)|(0<<ISC01)|(0<<ISC00)
out  MCUCR,tempa
cli

rjmp loop
```