```
#include "switches.h"

#define BOUNCE_DELAY 3125

unsigned int alarmSet;    // whether alarm is set
signed short alarmHours;  // alarm hour setting
signed short alarmMinutes;// alarm minutes setting

//--------------------switchInit--------------------
// arm external interrupts for PP1-PP6
// Input: none
// Output: none
void switchInit(void) {
  alarmSet = 0;
  alarmHours = 0;
  alarmMinutes = 0;

  DDRP &= ~0x7E;  // sets PP1-PP6 as inputs from switches
  PIEP |= 0x7E;   // enables external interrupts for PP1-PP6
  PPSP &= ~0x7E;  // sets polarity to falling edge interrupts
  PIFP = 0x7E;    // acknowledges all flags to prevent an immediate interrupt
}

interrupt 56 void switchHandler() {
  unsigned static long startTime = 0;
  if(TCNT - startTime > BOUNCE_DELAY) { // debouncing
    if(alarmOn) {   // any button turns off alarm if alarm is sounding
      alarmOn = 0;
      alarmSet = 0;
      PIFP = 0x7E;
    }
    else {          // otherwise, checks which button was let go
      if(PIFP & 0x02) { // toggles whether alarm is set
        alarmSet = ~alarmSet;
        PIFP = 0x02;    // acknowledges flag
      }
      if(PIFP & 0x04) {
        if(PTP & 0x40) {  // increments alarm minutes if PT6 is pressed
          alarmMinutes++;
        }
        else {            // otherwise, increments clock minutes and resets seconds
          seconds = 0;
          minutes++;
        }
        PIFP = 0x04;      // acknowledges flag
      }
      if(PIFP & 0x08) {
        if(PTP & 0x40) {
          alarmMinutes--; // decrements alarm minutes if PT6 is pressed
        }
        else {
          seconds = 0;
          minutes--;      // otherwise, decrements clock minutes and resets seconds
        }
        PIFP = 0x08;      // acknowledges flag
      }
      if(PIFP & 0x10) {
        if(PTP & 0x40) {  // increments alarm hours if PT6 is pressed
          alarmHours++;
        }
        else {            // otherwise, increments clock hours and resets seconds
          seconds = 0;
          hours++;
        }
        PIFP = 0x10;      // acknowledges flag
      }
      if(PIFP & 0x20) {
        if(PTP & 0x40) {
          alarmHours--;   // decrements alarm hours if PT6 is pressed
        }
        else {
          seconds = 0;
          hours--;        // otherwise, increments clock hours and resets seconds
        }
        PIFP = 0x20;      // acknowledges flag
      }
      if(PIFP & 0x40) {   // interrupt doesn't change anything
                          // only used to stop alarm and change alarm time
        PIFP = 0x40;      // acknowledges flag
      }

      // corrects if hours or minutes for either clock or alarm
      // go out of bounds
      if(hours >= 24) {
        hours = 0;
      }
      if(hours < 0) {
        hours = 23;
      }
      if(minutes >= 60) {
        minutes = 0;
      }
      if(minutes < 0) {
        minutes = 59;
      }

      if(alarmHours >= 24) {
        alarmHours = 0;
      }
      if(alarmHours < 0) {
        alarmHours = 23;
      }
      if(alarmMinutes >= 60) {
        alarmMinutes = 0;
      }
      if(alarmMinutes < 0) {
        alarmMinutes = 59;
      }
    }
    startTime = TCNT; // stores time for debouncing
  }
  else {              // only acknowledges flag if debounce delay hasn't
    PIFP = 0x7E;      // been met
  }
}
```