```
  ET1   = 1;
  EA    = 1;
  TR0   = 1;
  TR1   = 1;
  // change priorities so timer 1 is highest.
  PS  = 0;
  PT1 = 1;
  PX1 = 0;
  PT0 = 0;
  PX0 = 0;
  /// @brief P0 is 7 segment LED driver
  P0  = segmentArray[0];
  /// @brief P1 is the seconds binary leds, DOT LED, and alarm led outputs
  P1  = 0xBF;
  /// @brief P2 is the digit select control
  P2  = 0x00;
  /// @brief P3 is the switch input, and counter input for the seconds clock (2 Hz).
  P3  = 0x3F;

  waitForTimeSet();

  // loop forever
  for(;;)
  {
    // if the previous digit select is not equal to the current digit select, update display.
    if(prev_digitSelect != digitSelect)
    {
      // Turn off the LED's for a moment, this reduces flicker issues.
      P0 = 0;

      // seconds, complimented since 0 is 1 or on.
      P1 = (P1 & 0xC0) | (!SET_A_SWITCH ? 0x00 : (~seconds & 0x3F));

      // update previous digit select
      prev_digitSelect = digitSelect;

      // assert digit select and set alarm tone every other seconds.
      P2 = (alarm_tone << 4) | (digitSelect & 0x0F);

      // turn the DOT LED on when seconds is 1, off when 0.
      DOT_LED = ((!SET_T_SWITCH || !SET_A_SWITCH) ? 0 : seconds & 0x01);

      // based on selected digit, send out the digit to the proper 7 segment led. if alarm switch is
      held, show the alarm set time.
      switch(digitSelect)
      {
        case SEG_ONE_MINUTE:
          P0 = segmentArray[(SET_A_SWITCH ? gs_timeKeeper.one_minutes :
          gs_alarmKeeper.one_minutes)];
          break;
        case SEG_TEN_MINUTE:
          P0 = segmentArray[(SET_A_SWITCH ? gs_timeKeeper.ten_minutes :
          gs_alarmKeeper.ten_minutes)];
          break;
        case SEG_ONE_HOUR:
          P0 = segmentArray[(SET_A_SWITCH ? gs_timeKeeper.one_hours : gs_alarmKeeper.one_hours)];
          break;
        case SEG_TEN_HOUR:
          P0 = segmentArray[(SET_A_SWITCH ? gs_timeKeeper.ten_hours : gs_alarmKeeper.ten_hours)];
          break;
```