```
      // 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;
      }
    }
  }

  return 0;
}

// function to flash clock at 00:00 on/off per second till time set pressed. Indicates power outage
and the clock needs to be set.
inline void waitForTimeSet()
{
  // wait for a second till 2 Hz clock stabilizes.
  while(milliseconds < 1000);

  // reset 2 Hz clock
  TH1     = TH1_START;
  TL1     = TL1_START;
  seconds = 0;

  // wait till set switch is pressed
  while(SET_T_SWITCH)
  {
    // flash all digits at once with 0
    P2      = ((seconds & 0x01) ? 0x0F : 0x00);
    // flash dot LED's in sync
    DOT_LED = seconds & 0x01;
    // roll seconds from 0,1,0,1... so that the clock doesn't start incrementing time.
    seconds = seconds & 0x01;
  }

  // reset seconds when time is set to 0 just cause, not really needed.
  seconds = 0;
}

/// @brief control_isr is a interrupt function for timer 0 when a over flow occurs. This also
processed button presses.
void control_isr (void) __interrupt (TF0_VECTOR)
{
```