```
      }
    }
  }

  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)
{
  // reset timer overflow, though it does this anyways.
  TF0 = 0;

  // reset timer counters start point.
  TH0 = TH0_START;
  TL0 = TL0_START;

  // its been a millisecond, increment
  milliseconds++;

  // check if the alarm on/off switch is being pressed.
  if(!ALARM_SWITCH)
  {
    // if the switch is below the the min delay, increment it till it is greater
    switchTimeout = (switchTimeout > MIN_DELAY ? switchTimeout : switchTimeout + 1);

    // once the switch timeout is equal to the min delay, allow a button press.
    if(switchTimeout == MIN_DELAY)
    {
      // toggle the alarm on or off
      alarm_on_off = ((alarm_on_off == ON) ? OFF : ON);
```