```
/// @brief Global variable to store the current tone set from clock divider to 4051 router.
volatile uint8_t  alarm_tone          = 0;

/// @brief 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();

/// @brief main entry point for program.
int main(void)
{
  /// @brief local variable to store previous digitSelect value. Only set ports when it changes to
keep application from resetting values needlessly.
  uint8_t prev_digitSelect = 1;

  // Setup 89s51 for timer 0, counter 1, and interrupt enable.
  TMOD  = 0x51;
  TH0   = TH0_START;
  TL0   = TL0_START;
  TH1   = TH1_START;
  TL1   = TL1_START;
  // enable interrupts
  ET0   = 1;
  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);
```