```
// 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);

    ALARM_LED = !alarm_on_off;

    // make sure to turn off the tone if the alarm is turned off.
    if(alarm_on_off == OFF)
    {
      prev_milliseconds = 0;
      alarm_tone = 0;
    }
  }
}
// check if the alarm set switch is being pressed.
else if(!SET_A_SWITCH)
{
  // increment switch timeout
  switchTimeout++;

  // when both switches are not pressed, reset initial delay.
  if(MINUTE_SWITCH && HOUR_SWITCH)
  {
    initTimeout = INIT_DELAY;
  }

  // when either switch is pressed, and the press as exceeded the current timeout allow a button
press
  if((!MINUTE_SWITCH || !HOUR_SWITCH) && (switchTimeout > initTimeout))
  {
    // when minute is pressed add one
    gs_alarmKeeper.one_minutes += (MINUTE_SWITCH ? 0 : 1);

    // when hour is pressed add one
    gs_alarmKeeper.one_hours += (HOUR_SWITCH ? 0 : 1);

    // the below is the same code used in timer ISR. copy pasta with tweaks
    if(gs_alarmKeeper.one_minutes > 9)
    {
      gs_alarmKeeper.ten_minutes++;
      gs_alarmKeeper.one_minutes = 0;
    }
```