

```

(setq *work-time-start* 8)
(setq *work-time-end* 16)

;(setq *work-day-dow-table* [nil (9 . (17 30)) (9 . (17 30)) (9 . (17 30)) (9 . (17 30)) (9 . (17 30)) (9 . (17 30)) (9 . (17 30))])

(defun wt-dow-update-start-end nil
  (let ((beg-end (aref *work-day-dow-table* (nth 6 (decode-time (current-time))))))
    (if beg-end (setq *work-time-start* (car beg-end) *work-time-end* (cdr beg-end))
      (setq *work-time-start* 0 *work-time-end* 0))))

(setq *work-time-statuslen* 16)

(defmacro def-extract-if-list (name fun &optional ret)
  `(defun ,name (val)
    (if (listp val) (,fun val)
      , (if ret ret 'val))))

(def-extract-if-list wt-hour first)
(def-extract-if-list wt-minutes second 0)

(defun work-time-update nil
  (when *work-day-dow-table* (wt-dow-update-start-end))
  (let* ((now (current-time))
    (today (let ((dcd-now (decode-time now))
      ` (, (nth 3 dcd-now) , (nth 4 dcd-now) , (nth 5 dcd-now))))
    (start-time (apply #'encode-time (append (list 0 (wt-minutes *work-time-start*) (wt-hour *work-time-start*))
      (end-time (apply #'encode-time (append (list 0 (wt-minutes *work-time-end*) (wt-hour *work-time-end*))
      (labels ((status-string-format (perc)
        (cond ((< perc 0) ":(")
          ((> perc 1) "(:)"))
        (t (let* ((filled (round (* perc *work-time-statuslen*)))
          (empty (- *work-time-statuslen* filled)))
          (concat "[ (make-string filled ?#) (make-string empty ?#) "
            (elapsed-percentage (from to now) (/ (float (- now from)) (- to from)))
            (total-sec (time) (+ (* 65536 (car time)) (cadr time))))))
      (setq wt-status-string
        (status-string-format
          (elapsed-percentage (total-sec start-time) (total-sec end-time) (total-sec now))))))

(push #'work-time-update display-time-hook)

;

(defun wt-status-displayed nil
  (member 'wt-status-string global-mode-string))

(defun wt-add-status nil
  (setq global-mode-string (append global-mode-string '(wt-status-string))))

(defun wt-remove-status nil
  (setq global-mode-string (remove 'wt-status-string global-mode-string)))

(defun work-time-status-toggle nil
  (interactive)
  (funcall (if (wt-status-displayed) #'wt-remove-status #'wt-add-status)))

;

(setq wt-status-string (work-time-update))

```