

Re: Serial Port behavior in tcl

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2004-11/1185.html>

From: Rolf Schroedter (*me_at_privacy.net*)

Date: 11/29/04

Date: Mon, 29 Nov 2004 08:51:57 +0100

Have you thought about connecting the signal to the DSR,DCD or RI input signals ?
IMO that's more straight forward than using the RX pin.
Tcl since 8.4 supports polling these signals with
 fconfigure \$serial -tystatus

Rolf.

Andy McAllister wrote:

> I have an application which is very simple – I have created a
> peripheral sensor which outputs +7VDC on the RX pin on a laptop with
> tcl/tk 8.4.7. When the sensor detects a magnet (from a model train) it
> puts 0 VDC on the RX pin. I want to make a tcl app on the laptop that
> counts the number of times the sensor senses a magnet.
>
> I have 3 machines loaded with tcl 8.4.7. The sensor and my app works
> fine on both of these:
> – a Dell Latitude P3 laptop in Win2K
> – a P4 Panasonic laptop running redhat 8.0
> But my P133 Sharp laptop running Redhat 5.2 doesn't work with the
> sensor.
>
> Using the "term.tcl" application (code pasted below) I can test the
> port – I just connect pins 2 and 3 together to make a loopback. I can
> type and see my typing on screen, so I know the serial port works on
> the Sharp laptop. One thing I noticed, it does not work at 9600 baud,
> but does loopback chracters at 19600, 2400, and 1200 baud. So I know
> the port works and that /dev/ttyS0 is correct, and I know the sensor
> works to trigger fileevent on the other machines. But the sensor does
> not trigger a fileevent on the Sharp laptop like it does the Dell and
> Panasonic laptops. I put a bell into the receive subroutine, but never
> get a bell when I have the sensor plugged in and triggering on the
> Sharp laptop.
>
> Note that using the term.tcl program on the Panasonic laptop causes a
> symbol "x/00" to print on the terminal. I don't care what symbol is
> sent, just so I get SOMETHING in the laptop's RX serial buffer to
> trigger fileevent which will increment the counter. I have a timer in

comp.lang.tcl: Re: Serial Port behavior in tcl

```
> my code (included below as the file train.tcl, largely based on
> term.tcl) to assure only one trigger can happen in so many seconds.
>
> I know this is crude, but I wanted to keep it simple...
>
> If anyone has some insight, I'd greatly appreciate it.
>
> ## term.tcl:
> #####
> # Term for a simple terminal interface
>
> #####
> # The terminal bindings are implemented by defining a new bindtag
> 'Term'
>
> #####
> # Configure your serial port here
> #
> set Term(Port) com1
> set Term(Mode) "19200,n,8,1"
> set Term(Font) Courier
>
> # Global variables
> #
> set Term(Text) {}
>
> ##### Terminal In/Out events
> #####
> proc term_out { chan key } {
> switch -regexp -- $key {
> [\x07-\x08] -
> \x0D -
> [\x20-\x7E] { puts -nonewline $chan $key; return -code break }
> [\x01-\x06] -
> [\x09-\x0C] -
> [\x0E-\x1F] -
> \x7F { return }
> default { return }
> } ;# switch
>
>
> proc term_in { ch } {
> upvar #0 Term(Text) txt
>
> switch -regexp -- $ch {
> \x07 { bell }
> \x0A { # ignore }
> \x0D { $txt insert end "\n" }
> default { $txt insert end $ch }
> }
```

```

> $txt see end
> }
>
> proc receiver {chan} {
> foreach ch [ split [read $chan] {}] {
> term_in $ch
> }
> }
>
> ##### Windows #####
> proc scrolled_text { f args } {
> frame $f
> eval {text $f.text \
> -xscrollcommand [list $f.xscroll set] \
> -yscrollcommand [list $f.yscroll set]} $args
> scrollbar $f.xscroll -orient horizontal \
> -command [list $f.text xview]
> scrollbar $f.yscroll -orient vertical \
> -command [list $f.text yview]
> grid $f.text $f.yscroll -sticky news
> grid $f.xscroll -sticky news
> grid rowconfigure $f 0 -weight 1
> grid columnconfigure $f 0 -weight 1
> return $f.text
> }
>
> ##### main #####
>
> set chan [open $Term(Port) r+]
> fconfigure $chan -mode $Term(Mode) -translation binary -buffering none
> -blocking 0
> fileevent $chan readable [list receiver $chan]
>
> set Term(Text) [scrolled_text .t -width 80 -height 25 -font
> $Term(Font) ]
> pack .t -side top -fill both -expand true
>
> bind $Term(Text) <Any-Key> [list term_out $chan %A]
>
> catch {console hide}
>
>
>
> ##Train.tcl
> #####
> # Train for BagelME!
>
> #####
> # The terminal bindings are implemented by defining a new bindtag
> 'Term'
>

```

comp.lang.tcl: Re: Serial Port behavior in tcl

```
> #####
>
> # Configure your serial port here
> #
> set Term(Port) /dev/ttyS0
> set Term(Mode) "2400,n,8,1"
> set dist 0
> set trackdist 0.1
> # this is in inches for distance around the track
> set hysteresis 1.5
> # this var is in seconds to wait between signals on the sensor
> set lasttime [clock seconds]
>
> # Global variables
> #
> set Term(Text) {}
>
> ##### Terminal In/Out events
> #####
>
> proc term_in {} {
> global dist trackdist
>
> set dist [expr $dist + $trackdist]
>
> .count config -text $dist
> }
>
> proc receiver {chan} {
> global lasttime hysteresis
> if { [catch { gets $chan } fid] } {
> puts stdout "Serial got messed up: $fid"
> }
>
> if {[expr $lasttime + $hysteresis] < [clock seconds]} {
>
> term_in
> set lasttime [clock seconds]
> } else {
>
>
> }
>
> ##### Windows #####
>
> button .quit -text Quit -command exit
> set fonttype [font create Arial]
> font configure $fonttype -size 400
> label .count -text $dist -font $fonttype
> pack .quit .count -side bottom
```

comp.lang.tcl: Re: Serial Port behavior in tcl

```
>  
>  
>  
> ##### main #####  
>  
> set chan [open $Term(Port) r+]  
> fconfigure $chan -mode $Term(Mode) -translation binary -buffering none  
> -blocking 0  
> fileevent $chan readable [list receiver $chan]  
>  
>  
> catch {console hide}  
>
```

--

Rolf Schroedter, German Aerospace Center
Remove .nospam to reply: <mailto:Rolf.Schroedter@dlr.de.nospam>