AgilityContest 3.9.X

API del interfaz de comunicaciones serie para cronómetros con puerto RS232

Autor: Juan Antonio Martínez <info@agilitycontest.es>
Versión: 1.0 Mayo 2019

El siguiente documento indica el protocolo de comunicaciones que maneja el programa interfaz de comunicaciones serie que permite conectar cronómetros que dispongan de dicho interfaz a la aplicación AgilityContest

Contenido:

Descripción física del conector. Protocolos de bajo nivel

Líneas utilizadas (vistas desde el conector RS232 del PC)

- Requeridas
(5) GND masa común
(2) RxD (entrada) Recepción de datos en el PC
(3) TxD (salida) Envío de datos desde el PC

- Opcionales
(4) DTR (salida) Indica que el programa está activo y listo para enviar/recibir
(6) DSR (entrada) Indica que el cronómetro está activo y listo para enviar/recibir
(9) Ri (entrada) Indica activación de alguno de los sensores

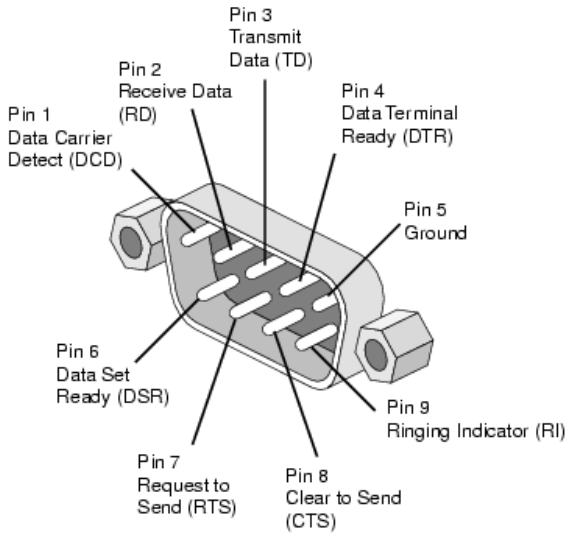
* Si el cronómetro no soporta el protocolo DTR/DSR, se recomienda que el cable tenga cortocircuitados los pines 4 y 6, o bien deshabilitar dicho protocolo en la aplicación de interfaz (opción por defecto)
* El uso de la línea Ring (pin 9) para indicar el paso por sensores NO debe usarse de manera simultánea con los comandos START/INT/STOP del protocolo

Conector DB-9

Descripción y pinout del conector DB9, usado en los PC's para el interfaz serie (RS-232).

Pin	SIG.	Signal Name	DTE (PC)
1	DCD	Data Carrier Detect	in
2	RXD	Receive Data	in
3	TXD	Transmit Data	out
4	DTR	Data Terminal Ready	out
5	GND	Signal Ground	-
6	DSR	Data Set Ready	in
7	RTS	Request to Send	out
8	CTS	Clear to Send	in
9	RI	Ring Indicator	in

El DTE (Ordenador) tiene el conector macho (mostrado abajo), y el DCE (Cronómetro) el conector hembra



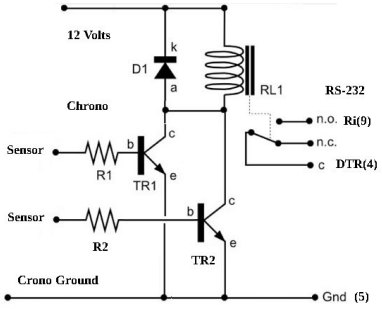
Protocolo de comunicaciones utilizado:

Data transmission: Asynchronous Serial
Speed: configurable (default 9600bps)
Partity: NONE
StopBits: 1
Synchronization: Full duplex
Handshaking: None (DTR/DSR opcional)
Signal levels: MARK = -3 to -15 V ... logic '1'
SPACE = +3 to +15 V ... logic '0'

Sugerencia de conexionado de los sensores de paso (opcional)

En el caso de que se desee usar el pin 9 (Ring) como indicador de paso, será necesario activar éste (ponerlo a nivel bajo) durante el tiempo en que el sensor detecta el corte del haz

El método recomendado es el uso de un relé u opto-acoplador que ante la activación del sensor conecte la señal DTR (pin 4) a la señal Ring(pin 9) durante el tiempo que el sensor está activado, tal y como se muestra en el esquema:



D1: 1N4004
R1 / R2: 1KΩ
TR1 / TR2: 2N2222
RL1: Relé 12 Voltios

En el esquema se asume que cuando el sensor está activado, en la línea “sensor input” aparece una tensión de 12 voltios

Recordar de nuevo que el uso de la linea Ring para el control de paso es opcional, y que en caso de implementarse NO DEBEN usarse los comandos START / INT / STOP del API de comunicaciones simultáneamente con dicha señal

Serial Communications Data Protocol (API)

API Command description:

- Is up to the user select serial port baudrate. No parity and 1 stop bit is assumed elsewhere
- Upper/Lower case is ignored
- Every command are ASCII text based, and ends with 0x0D 0x0A sequence ("windows/DOS newline")
- Extra whitespaces and non-ASCII characters are ignored
- Unknown commands or extra keywords should also be ignored
- Time stamps are given in miliseconds; but if dot/comma is present in timestamp, seconds are assumed
- Important: time deltas are *trunk'd* to lowest cents of second; NOT round'd, as KCC directives says

Messages from Chronometer to Computer:

- **START** [timestamp] < newline > (required)
  Chronometer starts. Timestamp mark is optional. When ommited zero (0) is assumed
- **INT** timestamp < newline > (optional)
  Intermediate course run timestamp. Time shown is "timestamp - start", trunk'd to cents of seconds
- **STOP** timestamp < newline > (required)
  End of course run. Time shown is "timestamp - start", trunk'd to cents of seconds
- **FAIL** < newline > (optional)
  Sensor failure. Should be sent every second while error remains
- **OK** < newline > (optional - required whenever fail is implemented)
  Chronometer is ready. Sensor error is over

Messages from Computer to Chronometer:

- **MSG** message [seconds] < newline > (optional)
  Show message on chronometer display . "seconds" is optional and indicates seconds that message is to be shown
- **TIME** [hh:mm] (optional)
  Tell chronometer to run in clock mode (that is, show HH:MM time). If time mark is ommited, chronometer should use their internal clock data. Receiving RESET will return to chronometer mode

Bi-Directional messages:

Can be sent either for the chronometer or the Computer.
Chronometer can ignore these commands, but honoring RESET is recommended

- **WALK** [seconds] <newline>
  Start course walk coundown. Seconds is optional, defaults to 420 (7minutes)
  Setting seconds to zero means stop course walk coundown
- **DOWN** [seconds] < newline >
  Start CountDown when competitor receives ack to run. Seconds is optional. defaults to 15 secs
- **FAULT** + < newline >
- **FAULT** - < newline >
- **FAULT** number < newline >
  Increase / Decrease / Set fault counter
- **REFUSAL** + < newline >
- **REFUSAL** - < newline >
- **REFUSAL** numero < newline >
  Increase / Decrease / Set refusal counter. If up to the user set ELIM flag after 3 refusals
- **ELIM** < newline >
- **ELIM** + < newline >
  Set eliminated mark
- **ELIM** - < newline >
  Clear eliminated mark
- **RESET** < newline >
  Clears chronometer status, setting zero fault/refusal/countdown counters. Stop and clears chronometer

AgilityContest_SerialChronometer

The AgilityContest plugin for for chronometers with serial port interface allows connect to AgilityContest Event bus any chronometer that uses serial port and complaint with these protocol

Provides:

- Command line invocation
- Configuration from command line options or .ini file
- Can also be started from AgilityContest console
- Web interface to show status. Can be used as auxiliary chronometer display
- Monitoring, test and logging features

Invocation:

ac_chrono[.exe] [options]

Where options are:
-d comport || --device=com_port
Communication port to attach to
-b baud || --baud=baudrate
Set baudrate for comm port. Defaults 9600 bauds.
-p tcpport || --port=tcp_port
Where to listen for web interface. Default 8080. When set to zero (0) web interface is disabled
-s ip || --server=ip_address
Location of AgilityContest server. Default "localhost"
-r ring || --ring=ring_number
Tell server which ring to attach chrono. Default "1"
-l level || --debuglog=level
Set debug/logging level from 1 (“none”) to 8 (“all”) . Defaults to 3 ("error")
-f file || --logfile=filename
Set log file. Defaults to "null" (“stderr” when in no-daemon mode)
–no-daemon
Test mode. Keep console open, to allow direct commands being typed in (default: daemon mode)
--find-ports
Do not run program, just show available , non-busy comm ports
--hw-prot
Use DTR/DSR serial lines to enable/disable comunications (default off)
NOTE: Future versions of the program may interpret that DSR inactive means sensor failure
--hw-sensor
Use Ring serial line as sensor activation signal (default off). Remember that this option
is incompatible with simultaneous use of START/INT/STOP API commands

In AgilityContest version 3.9.2 ac_chrono can also be launched from session control dialog.

Once is started, user can connect to ac_chrono via their web interface. In this way user can control most of functions and commands from api. Web interface can also be used as auxiliar chrono display

For AgilityContest Event bus API please read proper documentation.