Tracking System Server – Protokollversion 1.8

Floris Ernst, Matthias Knöpke

25. Oktober 2010

Inhaltsverzeichnis

1 Einleitung

In diesem Dokument wird das Kommunikationsprotokoll für die Tracking Server des Instituts für Robotik beschrieben.

2 Anmeldevorgang

Voraussetzung für die Anmeldung eines Clients ist der gestartete Trackingserver (Bsp.: AtracsysServer::main). Der Startvorgang beinhaltet die Initialisierung des Trackingsystems (Bsp.: AtracsysServer::createTrackingSystem) und des Trackingservers (Server::init). Nach erfolgreichem Starten wartet der Trackingserver auf Clients (Server::waitForClients).

2.1 Anmeldung eines Clients am Server

2.1.1 Verbindungsaufbau mit dem Trackingserver (Server::waitForClients)

Über TCP-Sockets wird eine Verbindung zum Trackingserver aufgebaut. Standardmäßig verbinden sich Clients auf Port 5000, für jeden Client wird ein eigener ServerThread gestartet.

Bespiel:

start Telnet 141.83.19.144 5000
receive Trying 141.83.19.144...
receive Connected to 141.83.19.144.

2.1.2 Initiale Kommunikation zwischen Server und Client

Die Folgenden drei Schritte müssen durchlaufen werden.

1. Beginn der Initialisierung
   Die Initialisierung erwartet vom Client das Kommando „CM_GETSYSTEM“. Das Trackingsystem antwortet mit einer Zeichenfolge. Diese besteht aus ANS_TRUE und Key-Value-Paaren („key=value“). Diese sind durch Leerzeichen getrennt.

   Bespiel:

   send CM_GETSYSTEM
   receive ANS_TRUE Protocol=1.3 Revision=1.1 Tracker=CameraCube;ArminGeom1;USPointer

Wie im Beispiel zusehen ist, werden Status, Protokoll, Revision und verfügbare Marker angezeigt. Sie werden in aktive, passive und virtuelle Tracker und durch einen Namen unterschieden.
Passive Marker werden auch Wireless-Marker genannt, sie arbeiten ohne Verbindung zum Trackingsystem. Micron, Vicra und Polaris gehören dazu.
Ein aktives Trackingsystem hingegen steuert seine Marker an. Aurora, MicroBIRD, Flashpoint, accuTrack und Polaris zählen dazu.
Virtuelle Marker gehören zu den aktiven Markern. Sie wurden für das atracsys accuTrack-System eingeführt. Ein aktiver Marker besteht aus vielen LEDs (eine jede ist ein virtueller Marker). Jeder Punkt kann einzeln detektiert werden. Es beschränken sich die Informationen jedoch auf die x-, y- und z-Koordinaten.

2. Auswahl eines Trackers (Markers)
   Im Anschluss ist der für den Client interessante Tracker (Marker) zu wählen. Sofern keine Fehler aufgetreten sind, wird der Tracker im Trackingserver aktiviert.

   Bespiel:

   send ArminGeom1
   receive ANS_TRUE

3. Festlegen des Ausgabeformats
   Für die Ausgabe der Trackingdaten stehen verschiedene Formate zur Verfügung (siehe Kap. 2.3). Die Wahl eines geeigneten Formats muss nun getätigt werden.

   Bespiel:

   send FORMAT_QUATERNIONS
   receive true

2.2 Befehlsmode

Nun stehen dem Client alle Funktionen des Trackingservers zur Verügung.

Bespiel:

send CM_NEXTVALUE
receive 1217942436.0150 y 0.604512 -0.296772 -0.338627 0.657132 -314.2929 -65.7137 -1840.4316 0.1903

2.3 Ausgabeformate der Trackerdaten

Bei der Initialisierung des Clients am Trackingserver wird das Ausgabeformat festgelegt. Die Kommandos CM_NEXTVALUE und CM_GETVALUEAT verwenden die eingestellte Formatierung. Der Formatparameter setzt sich aus drei Teilen zusammen.

FORMAT $formattyp[option]$

formattyp	_QUATERNIONS	
	_MATRIXROWWISE	
	_FORCETORQUE	
option	_M	Markermode
	_FRAMES	Framenumber (default Timestamp)
	_M_FRAMES	Markermode und Framenumber

RETURN $\{t\}\{vis\}\{a[..]\}\{q\}[Markermode][AddInfo]$

t	Timestamp oder Framenumber
vis	Sichtbarkeits-Flag, standardmäßig „y“ oder „n“. Mehr Optionen siehe CM_SETVISMODE.
$a[..]$	Formattyp QUATERNIONS: 4x Quaternions und 3x Translationen Formattyp MATRIXROWISE: 12x Werte Formattyp FORCETORQUE: 3x Kräfte und 3x Drehmomente
q	Qualität (wenn verfügbar, sonst -1)
$Markermode$	$\{3n\}\{x_1\ y_1\ z_1 \ldots\ x_n\ y_n\ z_n\}$ n - Anzahl der Markerkugeln Gibt die Koordinaten aller Markerkugeln getrennt zurück Ist nur bei Polaris implementiert/sinnvoll
$AddInfo$	Kann für weitere Trackinginformationen genutzt werden

Achtung: Die Optionen für den Formattyp sind nicht mit dem Formattyp FORCETORQUE kombinierbar!

BEISPIEL:

send	FORMAT_QUATERNIONS_M	
receive	ANS_TRUE	
send	CM_NEXTVALUE	
receive	1219149250.390000	\\ Timestamp
	y	\\ Sichtbarkeit
	0.98607534 0.16327722 -0.01786013 -0.02601887	\\ Quaternionen
	-11.865524 -32.695183 -819.444885	\\ Translationen
	0.500761	\\ Qualität
	15	\\ Trackeranzahl x3
	-170.100000 -11.230000 -833.640000	\\ x_1 y_1 z_1
	-239.560000 -1.760000 -840.820000	\\ x_2 y_2 z_2
	-268.540000 -37.510000 -860.330000	\\ x_3 y_3 z_3
	-268.500000 42.400000 -8 26.830000	\\ x_4 y_4 z_4
	-313.210000 7.960000 -847.560000	\\ x_5 y_5 z_5

2.4 Antworten vom Trackingserver

- ANS_TRUE
  Kommando wurde ausgeführt. Wenn Werte zurückgeliefert werden, dann kein ANS_TRUE
- ANS_FALSE
  Kommando konnte nicht umgesetzt werden (z.B. unbekannt oder falsche Parameter), Kommando selbst aber bekannt
- ANS_UNKNOWN *cmd*
  Server kennt Befehl *cmd* nicht
- PONG
  Antwort auf das Kommando CM_PING

2.5 Befehlsübersicht

- CM_GETSYSTEM
  Ist ein spezieller Befehl zur Anmeldungs des Clients am Trackeringservers. Der Rückgabewert setzt sich aus Antwort, Protocol, Revision, Tracker, Name (des Trackingsystems), Serial(Seriennummer), Firmware und Platform(Windows oder Linux) zusammen. Es handelt sich bei der Zeichenkette um Key-Value Paare (durch Leerzeichen getrennt).
  Beispiel:

send	CM_GETSYSTEM
receive	ANS_TRUE Protocol=1.2 Revision=1.1 Tracker=SofamorDanekNew;CameraCube
	Name=PolarisVicra Serial=P6-00167 Firmware=008 Platform=Windows

- CM_SETAVGMODE *mode*,*windowsize*

mode	AVERAGE	\\ Mittelwert
	WEIGHTEDSUM	
	EXPSMOOTHING	
windowsize	$\in \mathbb{N}$	

  Festlegen und Einstellen eines Mittelungsmodus. Zur Wahl stehen arithmetische, linear ansteigende und exponentielle Mittelung. Die Summe der Gewichtung über die windowsize ist 1. Bei der linearen und exponentiellen Mittelung sind die letzten Werte am höchsten gewichtet.
  Deaktiviert wird der Mittelungsmodus durch $windowsize = 0$.
- CM_SETVISMODE n
  Einstellen des Übertragungsmodus des visibility-Flags.

n	Beschreibung
0	Standard-Ausgabe: Sichtbarkeitsflag wird als „y“ oder „n“ ausgegeben
1	Erweiterte Ausgabe: Zusätzlich ist noch die Ausgabe „w“ möglich (Marker sichtbar, aber mit Warnungen)
2	Erweiterte Ausgabe, numerisch. „y“ entspricht 1, „n“ entspricht 0, „w“ entspricht 2.

- CM_NEXTVALUE
  Anfordern des letzten gespeicherten Trackingwertes. Abfragehäufigkeit und eingestellte Trackingfrequenz sind unabhängig voneinander.

- CM_NEXTVALUE_BLOCK
  Anfordern des letzten, nicht gesendeten Trackingwertes. Rückgabe des Servers erfolgt sofort, falls der letzte verfügbare Trackingwert noch nicht gesendet wurde, sonst, sobald ein neuer Wert verfügbar ist.

- CM_SETPUSHVALUES ON|OFF
  Durch dieses Kommando kann die momentane Verbindung vom Polling- in den Push-Modus umgeschaltet werden. Im Polling-Modus (CM_SETPUSHVALUES OFF, Standard) müssen neue Messwerte mit CM_NEXTVALUE oder CM_NEXTVALUE_BLOCK abgefragt werden. Im Push-Modus (CM_SETPUSHVALUES ON) werden neue Messwerte automatisch gesendet.

- CM_KILLSERVER
  Ermöglicht das Beenden des Trackingservers vom Client aus (unabhängig ob der Server als Standaloneprogramm oder dynamische Bibliothek gestartet wurde). Anwendungsfälle können sein, dass Server und Client zusammen stoppen oder Server neu gestartet werden muss.

- CM_GETVALUEAT *timestamp*
  timestamp - Unix-Timestamp
  Anfordern eines Wertes zum Zeitpunkt *timestamp*. Der Wert ist ein linear interpolierter Wert und wird zurückgegeben, sofern Werte vor und nach dem Zeitpunkt vorhanden sind. Bisher ist nur lineare Interpolation implementiert.

- CM_SETINTERPOLATION *mode*
  mode - LINEAR
  Festlegen bzw. Ändern des Interpolationsverfahrens welches für CM_GETVALUEAT verwendet wird. Bisher ist dieses Kommando überflüssig.

- CM_GETSTRAY
  Ist ein spezielles Kommando für Straymarker (des Polaris Trackingsystems). Ist im Trackingserver ein passiver Marker aktiv (und von mindestens einem Client ausgewählt), so werden mit diesem Befehl alle Positionen jener Markerkugeln ausgegeben, die nicht zu dem ausgewählten Tracker gehören. Es werden alle x, y & z Koordinaten nacheinander ausgegeben.
  Beispiel:

send	CM_GETSTRAY	
receive	1219149276.078	\\ Timestamp
	-25.1780987 22.5403805 -644.6302490	\\ Unbekannte Markerkugel1
	-12.6931601 -27.1938839 -643.0692139	\\ Unbekannte Markerkugel2
	63.9880104 -1.9875927 -631.4852905	\\ Unbekannte Markerkugel3

- CM_GETTRACKERS
  Liefert die Namen aller vom Server erkannten Tracker zurück (auch während der Initialisierung möglich).

- CM_GETTRACKERINFO *name*
  Liefert Informationen über den angefragten Tracker (auch während der Initialisierung möglich).
  Falls der mit *name* angegebene Tracker bekannt ist, ist die Antwort
  $type$,$init$,$connect$,$numVirtual$
  sonst ist die Antwort
  ANS_FALSE
  Hierbei ist *type* eines von *a* (aktiver Tracker), *p* (passiver Tracker), *v* (virtueller Tracker), *f* (Kraft-Momenten-Sensor), *s* (Synchronisations-Tool), *t* (anderer Tracker), *l* (einzelne LED), oder *u* (unbekannt). *connect* und *init* geben Auskunft darüber, ob ein Tracker angeschlossen (*connect*) und initialisiert (*init*) ist. $numVirtual$ ist die Anzahl der dem angefragten Tracker zugeordneten virtuellen Tracker.
  Beispiel (NDI Polaris):

send	CM_GETTRACKERINFO PolarisActive_1
receive	a,1,1,0

- CM_GETNUMVIRTUAL *name*
  Liefert die Anzahl der dem Tracker *name* zugeordneten virtuellen Tracker zurück (auch während der Initialisierung möglich).
  Beispiel (atracsys accuTrack):

send	CM_GETNUMVIRTUAL Boomerang
receive	4

- CM_PING
  Antwortet mit PONG. Das Kommando kann dazu benutzt werden, um die Antwortzeit des Systems zu bestimmen.

- CM_SETLOGLEVEL *mode*
  Festlegen des Loggingmodes für den gesamten Trackingserver.

mode	LOGLEVEL_QUIET	Keine Meldungen
	LOGLEVEL_ERROR	Fehlermeldungen
	LOGLEVEL_WARN	Warnmeldungen inkl. LOGLEVEL_ERROR
	LOGLEVEL_INFO	allgemeine Informationen inkl. LOGLEVEL_WARN
	LOGLEVEL_DEBUG	DebugMeldungen inkl. LOGLEVEL_INFO
		...werden ausgegeben

- CM_GETREVISION
  Anfordern der Revisionsnummer des Trackingservers.

- CM_QUITCONNECTION
  Verbindung zum Trackingserver beenden. Der verwendete Trackingthread wird beendet.

- CM_SETADDINFO *mode*
  mode - on | off
  Zusatzinformationen werden am Ende der Daten mitgeliefert. ADD_INFO wird nur geliefert, wenn *mode* == on gesetzt wird. Achtung: Kein Antwortstring!

- CM_GETSTROBEMODE
  Gibt den momentanen LPT-Strobing-Modus zurück:

  -1 Strobing nicht verfügbar
  0 Strobing verfügbar, aber deaktiviert
  1 Flanken-Strobe
  2 Peak-Strobe
  3 Up-Down-Strobe

- CM_SETSTROBEMODE
  Setzt den LPT-Strobing-Modus. Mögliche Werte sind:

  0 Strobing deaktivieren
  1 Flanken-Strobe
  2 Peak-Strobe
  3 Up-Down-Strobe

  Antwort mit ANS_TRUE, wenn der Modus geändert wurde, mit ANS_FALSE, wenn Strobing nicht verfügbar ist oder der Modus nicht geändert werden konnte.
  Beim LPT-Strobing wird über die Datenleitungen des LPT-Ports nach jeder erfolgten Abfrage des Trackingsystems ein Signal gesendet. Hierbei sind die Werte auf den Datenleitungen 1 bis 7 zufällig. Der Wert auf der Datenleitung 0 hängt vom Modus ab:

Flanken-Strobe	der Wert wechselt nach jeder Messung, d.h. es entsteht eine Rechteck-Funktion.
Peak-Strobe	nach jeder Messung wird die Leitung auf 1 gesetzt und direkt danach wieder auf 0, d.h. es entsteht eine Kamm-Funktion.
Up-Down-Strobe	es wird eine Dreiecksfunktion mit zufälliger und wechselnder Amplitude gefahren.

  Der Zahlenwert, der über die Datenleitungen gesendet wird, wird im Additional-Info-Feld gespeichert.

- CM_GETSTROBEVALUE
  Gibt den letzten gesendeten Strobe-Wert (zwischen 0 und 255) zurück

Protokollhistorie

1.8	Neue Kommandos: CM_SETVISMODE, CM_SETPUSHVALUES
	Neue Funktionalität: Synchronisationsmarker und einzelne LEDs (nicht-virtuell)
	Neue Formate s, l und t in CM_GETTRACKERINFO
1.7	Neue Kommandos: CM_GETSTROBEVALUE
	Neuer Strobemodus: up/down (mode = 3)
1.6	Neue Kommandos: CM_SETSTROBEMODE, CM_GETSTROBEMODE
	Neues Datenformat: FORMAT_FORCETORQUE
	Neues Format f in CM_GETTRACKERINFO
1.5	Neue Kommandos: CM_NEXTVALUE_BLOCK

1.4 Neue Kommandos: CM_PING
1.3 Neue Kommandos: CM_GETTRACKERINFO, CM_GETNUMVIRTUAL
Revision-Antwort geändert
1.2 Erweiterte Antwort auf CM_GETSYSTEM
1.1 Neue Kommandos: CM_GETTRACKERS, CM_GETREVISION
1.0 Initiale Version