Keysens

Telegrams

Technical note

Document title: technical note - Telegrams
Document number: 1001
Revision: 03
Date: June 2017

Copyright *Keysens*
Specifications are subject to change due to technical developments. Details presented may be modified.
All rights reserved.

Keysens - Machine Vision
Poligono Ecce Homo nave 4. 12530 Burriana, Spain
info@keysens.com
www.keysens.com

Contents

1 Introduction

Keysens vision *runtime* software listens as a server or connects as a client with one robot and one HMI per camera. The protocol is TCP/IP at the specified IP addresses and port numbers.

The connections are as a server or as a client depending on how this is configured in the configuration file *port.cfg*.

If the *runtime* acts as a server, the IP address is the one set in the Operating System and the port number the one set int the corresponding vision project. If it acts as a client, the IP adresses and port numbers are configured in the vision project of each camera.

Besides, the *runtime* opens a server connection used to send and receive vision projects from a configuration PC running the configuration program *vDevelop*, that acts as a client. This connection is made at the IP set in the Operating Systems and the port number set in the configuration file *port.cfg*.

This technical note refers only to the messages or telegrams received by the *runtime* from the robots or HMI. The communication from the configuration program *vDevelop* follows another format and is beyond the scope of this document.

Through the communication with robots and HMIs, the *runtime* sends results in several formats, sends a periodic heart beat message if configured, and accepts commands to perform several actions, such as changing the current projects or the value of any algoritm parameter.

The accepted commands perform these actions:

- Put the *runtime* in *stand by* state, in this state no image processing is performed.
- Put the *runtime* in *run* state, in this state processing is performed normally.
- Select a camera number so that following commands will refer to that camera.
- Load a new project identified by a project number, the project is saved to disk as the current project, it will be loaded the next time the *runtime* starts.
- Change a parameter value of an algorithm.
- Store the current project to disk, thus saving changes in algorithm parameters.

As mentioned, this technical note describes the format of the telegrams sent between the *runtime* and a robot or an HMI. These telegrams are divided into:

- Results of the image processing sent by the *runtime*.
- Configuration commands received by the *runtime*.

2 Telegrams format

All telegrams are coded in ASCII and enclosed in square brackets. That is, the first character is '`[`' and the last character is '`]`'. Command telegrams have a mnemonic of three characters following the first character '`[`'.

Numbers can be integer or real, positive or negative. In this document integer numbers are denoted by `999`, they can be of any length. Real numbers are denoted by `999.999`, the integer part can be of any length and the decimal part is of the indicated number of decimals (ex.: `999.99` is a real number with two decimals). If a number is negative it should start with character '`-`'. Numbers may have any leading blank spaces.

3 Results telegrams

The result of the image processing for each camera is a numeric matrix of a number or rows and columns. Usually, each row represent measured characteristics of a detected object. Each row of the matrix is sent as a telegram following the configured protocol format. The protocols used for sending results to the robots and to the HMIs are configured in the vision projects. Differnt protocols or the same protocol can be configured for the robot and the HMI assigned to a camera.

Currently there are two protocols implemented:

- Protocol *Keysens*.
- Protocol *Keba*.

Protocol *Keba* is named after the protocol specifications of *Keba* for its product *KeMotion*, a robot motion controller. *Keba* robots expect results from cameras in this protocol.

3.1 *Keysens* results protocol

The row numbers are set to two decimal figures. Number are separated by commas: '`,`'.

Ex.: a telegram with four numbers is:

`[999.99,999.99,999.99,999.99]`

The general format is:

`[999.99...]`

Where '`...`' means any more numbers with the same format, separated by commas: '`,`'.

3.2 *Keba* results protocol

In *Keba* protocol at most five numbers can be sent in a telegram. So, at most, only the first five numbers of a row are sent. If a row has more numbers, the rest are ignored.

The first three row numbers represent an object position and orientation. The fourth number represents an object attribute or object class, and the fifth one represents the object identificacion.

The first three row numbers are set to three decimal figures, the following two numbers are set to integers. Numbers are preceded by a label and a colon '`:`'. If there are more that one number per row, they are separated by a semicolon '`;`'. If a row has less than five numbers, only the present numbers are sent.

The format is:

`[X:999.999;Y:999.999;A:999.999;ATTR:999;ID:999]`

Each number has a different meaning, thus the labels are different. The meanings are: x position of the object centre, y position, angle of orientation, attribute and identification.

3.3 Heart beat

A heart beat message is sent to the robots and to the HMIs if configured in the corresponding vision project. This message indicates that the *runtime* is running normaly. It is a periodic message of period 2 seconds approximately.

The format of the heart beat telegram is:

```
[H]
```

4 Command telegrams

Configuration telegrams are received by the *runtime* through client or server connections with the robots and HMIs.

Currently, the following commands are implemented:

Telegram	Meaning
[ALG999,999,999]	Change a parameter value of an algorithm. First number is the algorithm order number starting from 1. Second number is the parameter order number starting from 1. Third number is the new parameter value.
[CAM999]	Select a camera number, future message commands will refer to this camera number.
[PRO999]	Change project, load the specified project number and store this project for the next session.
[RUN]	Put the *runtime* in *run* state, processing is performed normally.
[STB]	Put the *runtime* in *stand by* state, no image processing is performed.
[STO]	Store current project thus saving changes made to algorithm parameters.
[NUM]	Ask for the project number executed for the selected camera (select camera with telegram [CAM999]). Selected camera is 1 when program starts. The program responds with a telegram [PRO999], where 999 is the project number.

Numeric parameters are trimmed with a minimum and a maximum value, so that their value remains consistent with what is expected in the command.

The minimum and maximum values are:

Telegram	Minimum value	Maximum value
[CAM999]		
First parameter	1	3
[PRO999]		
First parameter	1	999
[ALG999,999,999]		
First parameter	1	max number of algorithms in project
Second parameter	1	max number of parameters in algorithm
Third parameter	min algorithm parameter value	max algorithm parameter value

5 Comments

If you experience any problems with this document or want to give us feedback, please email us at info@keysens.com.