

KAS

Interface App via variable remote access using http request Programmed with Android Studio



Record of Document Revisions

<i>Revision</i>	<i>Remarks</i>
A 03/2017	First version

IMPORTANT NOTICE

Registered Trademarks

PDMM and KAS are registered trademarks of Kollmorgen Corporation

Android is a registered trademark of the Google Inc.

Java is a registered trademark of the Sun Microsystems Inc.

Content

KAS_Iface.apk	Installation ready application file
HttpRequestTask.java	Asynchronous communication function
MainActivity.java	Main code of the application
activity_main.xml	Graphical user interface description
MN_KASIfaceApp_0A0_de	Manual in German
MN_KASIfaceApp_0A0_en	Manual in English

Contents

Introduction	4
<i>Syntax</i>	<i>5</i>
Installation	6
<i>KAS</i>	<i>6</i>
<i>KASIface</i>	<i>7</i>
Java Code	8
<i>MainActivity.java</i>	<i>8</i>
<i>activity_main.xml</i>	<i>8</i>
<i>HttpRequestTask.java</i>	<i>9</i>

Introduction

The *KASIface* application is to show how a KAS-variable access can be implemented into a mobile device. This example uses http request in the text format. The programming is done with Java in Android Studio.

KASIface

Kollmorgen Automation Suite

Control Panel

☐ Automatic

☐ Manual

☐ Off

E-STOP

Manual Mode Functions

TravelSpeed

[deg/s]

ABS MOVE

[deg]

REL MOVE

[deg]

Auto Mode Functions

Machine Speed

[deg/s]



User defined Variable access

Variable name

READ VALUE

Variable value

DISCLAIMER:

	<p>All programs in this release (application demos, Kollmorgen UDFBs, etc.) are provided “AS IS, WHERE IS”, WITHOUT ANY WARRANTIES, EXPRESS OR IMPLIED. There may be technical or editorial omissions in the programs and their specifications. These programs are provided solely for user application development and user assumes all responsibility for their use.</p> <p>Programs and their content are subject to change without notice.</p>
	<p>KOLLMORGEN srl cannot be held liable for any damage caused by the use of the “KASIface” or any parts of it.</p>

Syntax

The reading of a variable can easily be done in a web browser.

To read *Axis1Status* for example the following request is needed:

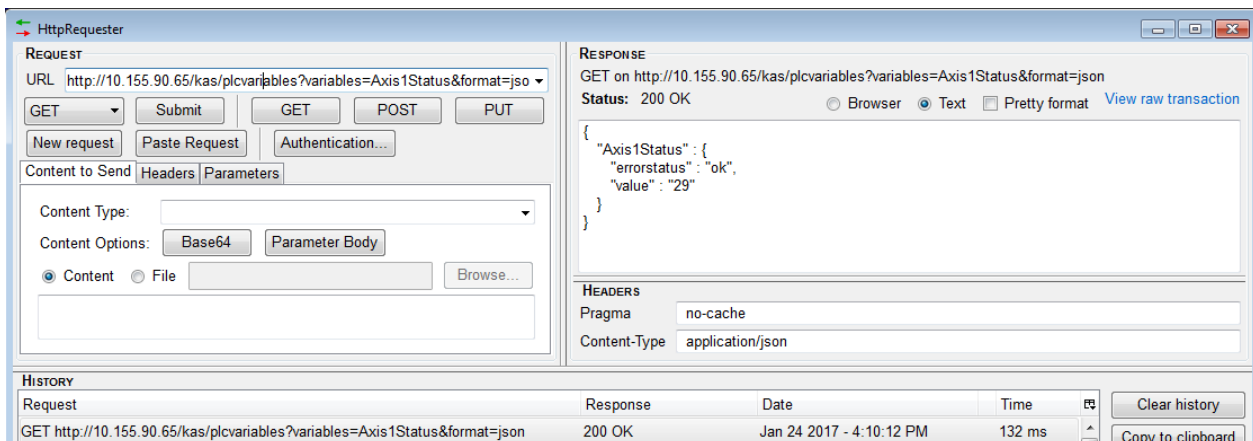
http://10.155.90.64/kas/plcvariables?variables=Axis1Status&format=text

To write a value of 1 to the variable *bEstop* the request is as following:

http://10.155.90.64/kas/plcvariables?&format=text
bEstop=1

Another example on the same topic done with excel or visual basic, and all the help files can be found on the Kollmorgen knowledge base [KDN](#).

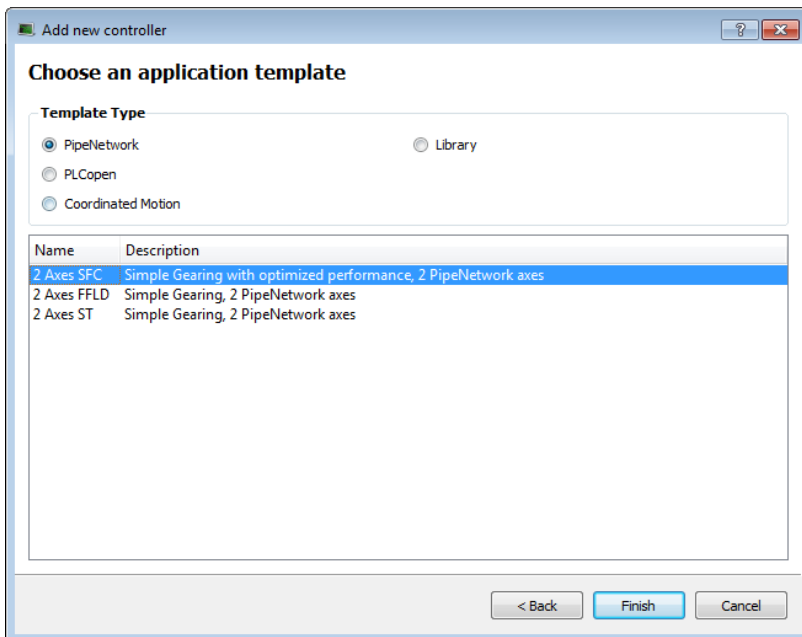
The firefox add-on HttpRequester gives an easy interface to test all the possible commands directly in the web browser.



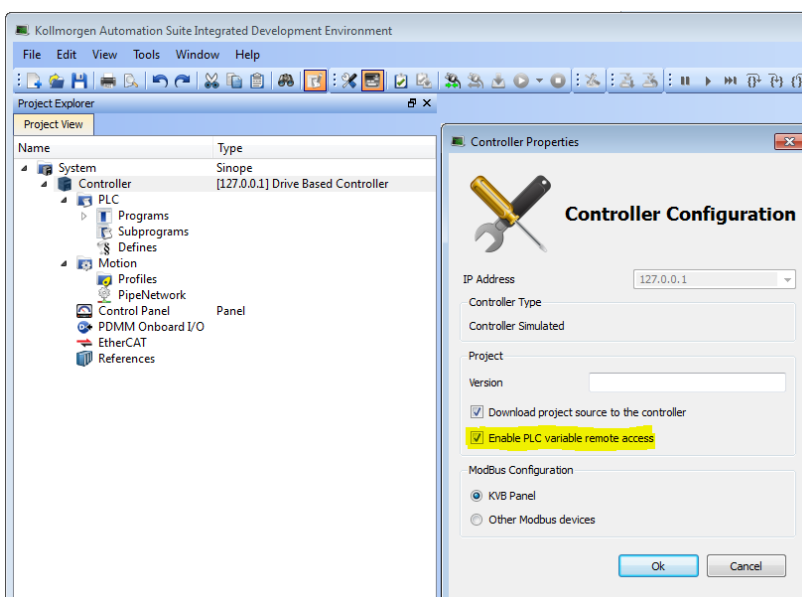
Installation

KAS

This example uses the SFC standard example for PipeNetwork defined in KAS.
File -> New -> PDMM (or corresponding controller) -> Next -> PipeNetwork -> 2 Axis SFC -> Finish. The Version for SFC has to be used, as with the FFLD and the ST example some commands and variable names are modified and not all tags will communicate.



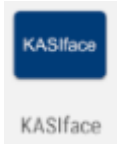
To use remote access to all KAS variable the function has to be enabled in the controller properties. This can be done with a right click on Controller and activating “Enable PLC variable remote access”.



With this setting changed the application can be compiled and loaded to the controller.

KASiface

KAS_iface can be installed directly on an android device. After installation, the following icon is added to the installed software screen. The software is now ready for use.



At startup, the IP-address of the KAS controller has to be entered. All communication is done with this IP-address. If communication is interrupted, all text boxes that read data show “no value”.

Enter the PxMM IP-Address

10.155.90.64

OK

After confirming the IP-address with OK all read variable are pulled cyclic (250ms). The writing is event based.

The following events are defined

Type	Application	Event
Button	E-Stop	At <i>click</i> the write toggles once between 0 and 1
	AbsMove, RelMove	At <i>click</i> a 1 is written which will be followed by a 0. This to always create a rising edge.
RadioButton	Mode	At <i>Statechange</i> the corresponding mode (0, 1, 2) is written
EditText	TravelSpeed, MasterAbsPos, MasterDeltaPos, MachineSpeed	At <i>click</i> the reading of the corresponding text field is stopped, until focus of the field is lost again. Writing is executed on <i>enter</i> . After writing, the focus is always lost automatically.
	User defined variable	If set to read the value field is constantly updated If set to write the value is only written on enter, never read

The application has been tested with a Samsung A3 and Android 6.0.1, there is no minimum version defined. Other Hardware or Software constellation may need changes, which are not listed here.

Java Code

There are three different pieces of java code related to this project added.

MainActivity.java

Main in which all basic functionality is located and all function calls are done.

protected void onCreate(Bundle)

- AlertDialog to enter the ip-address of the KAS controller
- Event handling for all writing
- Runnable Handler to cyclic call ReadKasVar()

public void WriteBoolTrueFalse(String, boolean)

- The Parameter with the name of *String* will be written with *boolean* (TRUE)
- After a successful call with *boolean* being true, automatically the write is executed with *boolean* being false. To reset the parameter back to zero.

public void ReadKasVar()

- All KAS variable to be read on cyclic base are added here
- This function is called cyclic all 250ms by the runnable handler

activity_main.xml

The xml contains all information about the user interface. The layout is done in relative mode with multiple grid layouts.

HttpRequestTask.java

Extension of the AsyncTask for all URL based requests. This function is called in the main activity every time a read or write to any KAS variable needs to be done.

The function can have up to three parameters:

First, the IP-address as string. In the example software, the IP-address is stored in *strIpApp*.

Second, the parameter name, as string.

Third, the parameter value to be written, as string.

The *AsyncResponse* contains the response of the text request as string. If a parameter is read this contains the read value. If a parameter is written this contains a "ok" if successful or the corresponding fault.

Lesen:

A call of the function with two parameters executes a read of the designated parameter.

```
AsyncTask<String,Void,String> ReadKasVar_AbsMove = new HttpRequestTask(new HttpRequestTask.AsyncResponse() {  
    @Override  
    public void AsyncResponse(String str_Result) {  
        if (!bWriteAbsMovePos) {  
            TextAbsMove.setText(str_Result);  
        }  
    }  
}).execute(strIpAddr, "MasterAbsPos");
```

Schreiben:

A call of the function with three parameters executes a write of the designated parameter.

```
AsyncTask<String,Void,String> WriteKasVar_AbsPos = new HttpRequestTask(new HttpRequestTask.AsyncResponse() {  
    @Override  
    public void AsyncResponse(String str_Result) {  
    }  
}).execute(strIpAddr, "MasterAbsPos", TempWriteValue);
```

About KOLLMORGEN

Kollmorgen is a leading provider of motion systems and components for machine builders. Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions that are unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.

For assistance with your application needs, visit www.kollmorgen.com or contact us at:

KOLLMORGEN srl

Largo Brughetti 1/B2

20813 Bovisio Masciago, Italia

Web www.kollmorgen.com

Email mil-info@kollmorgen.com

Tel.: +39 - 0362 - 594260

Fax: +39 - 0362 – 594263