Matthews MPERIA External Protocols Overview

for MPERIA 6.1.x

Table of Contents

1 - Revision history....3
2 - Overview....4
3 - Examples....5
3.1 - Set a single variable....5
3.2 - Set three variables....5

1 Overview

MPERIA supports several different protocols that can be used by remote systems. Each protocol has it's own strengths and weaknesses, which makes it important to pick the right protocol to use.

Line Action Protocol	Advantages: • Simple line based protocol. Just send the line of text. • The GUI is used to configure what the received line is used for, e.g. selecting a message or setting a variable. Disadvantages: • Can only do one thing per port, e.g. select a message, but a single port cannot set a variable and a counter. • Can only do simple things. No support for triggering, performing maintenance, reading status, etc.
XML Protocol	Is by default available on TCP/IP port 10001. Can be configured to be available on USB and serial ports. Advantages: • Based on the XML standard, which means that most high level systems have tools to generate and parse XML. • Supports advanced functions such as creating messages. • Can handle different character sets. Disadvantages: • Requires XML support.
Command Protocol	Is by default available on TCP/IP port 10005. Can be configured to be available on other TCP/IP ports and on USB and serial ports. Advantages: • Easy to use. • Extremely easy to parse. Responses are single bytes. • Low overhead. Disadvantages: • While more powerful than the Line Action Protocol, it cannot for example create messages.
Plugins	Can be configured to be available on TCP/IP ports and on USB and serial ports. Advantages: • Very flexible. • If the customer already have a system where they send the data in some format or another, they don't have to change it. Disadvantages: • There is a cost associated with developing the plugin.

2 Examples

2.1 Set a single variable

Via TCP/IP, set the value of the variable “PRODUCT” to “Bananas”.

Line Action Protocol	**Configuration:** • Add a TCP/IP port. • Select protocol: Line Action. • Create a new action: Set Variable 'PRODUCT'. **Send:** `Bananas`

XML Protocol	**Configuration:**
	None
	Send:
	`<request>`
	`  <setVariable name="PRODUCT">Bananas</setVariable>`
	`</request>`

Command Protocol	**Configuration:** None **Send:** `<ESC>VAR_SET<FS>PRODUCT<FS>Bananas<ETB>`

plugin_connection	**Configuration:**
	• Write the plugin.
	`function plugin_connection(connection, configuration)`
	`  local product = connection:readline()`
	`  mperia.variabel.set("PRODUCT", product)`
	`end`
	• Import the plugin.
	• Add a TCP/IP port.
	• Select protocol: Plugin.
	• Choose the plugin.
	Send:
	`Bananas`

plugin_action	**Configuration:**
	• Write the plugin.
	`function plugin_action(input, configuration)`
	`  mperia.variabel.set("PRODUCT", input)`
	`end`
	• Import the plugin.
	• Add a TCP/IP port.
	• Select protocol: Line Action.
	• Create a new action: Plugin 'myplugin.lua'.
	Send:
	`Bananas`

2.2 Set three variables

Via TCP/IP, set the value of the variable “PRODUCT” to “Bananas”, the variable “BATCH” to “42”and the variable “BOXSIZE” to “Large”.

Line Action Protocol	Not possible

XML Protocol	**Configuration:**
	None
	Send:
	`<request>`
	`  <setVariable name="PRODUCT">Bananas</setVariable>`
	`  <setVariable name="BATCH">42</setVariable>`
	`  <setVariable name="BOXSIZE">Large</setVariable>`
	`</request>`

Command Protocol	**Configuration:**
	None
	Send:
	`<ESC>VAR_SET<FS>PRODUCT<FS>Bananas<ETB>`
	`<ESC>VAR_SET<FS>BATCH<FS>42<ETB>`
	`<ESC>VAR_SET<FS>BOXSIZE<FS>Large<ETB>`

plugin_connection	**Configuration:**
	• Write the plugin.
	`function plugin_connection(connection, configuration)`
	`  local product = connection:readline()`
	`  local batch = connection:readline()`
	`  local boxsize = connection:readline()`
	`  mperia.variabel.set("PRODUCT", product)`
	`  mperia.variabel.set("BATCH", batch)`
	`  mperia.variabel.set("BOXSIZE", boxsize)`
	`end`
	• Import the plugin.
	• Add a TCP/IP port.
	• Select protocol: Plugin.
	• Choose the plugin.
	Send:
	`Bananas`
	`42`
	`Large`

plugin_action	**Configuration:**
	• Write the plugin.
	`function plugin_action(input, configuration)`
	`  local data = mperia.util.split(input, ";")`
	`  mperia.variabel.set("PRODUCT", data[0])`
	`  mperia.variabel.set("BATCH", data[1])`
	`  mperia.variabel.set("BOXSIZE", data[2])`
	`end`
	• Import the plugin.
	• Add a TCP/IP port.
	• Select protocol: Line Action.
	• Create a new action: Plugin.
	Send:
	`Bananas;42;Large`