IMV CORPORATION

SD logger script language tutorial format

Send

If the sent data is described after "/" or ":", it will be sent to the target.If the transmission data is described in text, put "/" at the beginning of the line, and if you want to describe it in hex (hexadecimal), put ":".
If "/" is placed at the beginning of the line, the characters from the next character to the line before the line feed code are sent, including spaces and tab codes. No line feed code is sent.
If you put ":" at the beginning of the line, it will be interpreted as a hexadecimal character and transmitted.
Hexadecimal characters can be uppercase or lowercase. Spaces and tabs are interpreted as delimiters and are not sent on their own. If there is no delimiter, it will be interpreted with two digits.

```
/ABC
```

In this example, "ABC (41 42 43)" is sent.

```
:01234 5 AbC
```

In this example, "(01 23 04 05 AB 0C)" is sent.

When sending a line feed code, do as follows.

```
/ABC
:0D
```

In this example, "ABC <CR> (41 42 43 0D)" is sent.

Send periodically

To send data repeatedly at regular intervals, write as follows.

```
#LOOP EVER
  #WAIT TIME 1S
  /ABC
#END
```

In this example, "ABC" is sent every second.

Wait for reception of specific data

To send specific data when receiving a specific data pattern, write as follows.

```
#LOOP EVER
  #WAIT DATA /XYZ
  /ABC
#END
```

In this example, when "XYZ" is received, "ABC" is sent.

To wait including the line feed code, write as follows.

```
#LOOP EVER
  #WAIT DATA /XYZ
  #WAIT DATA :0D
  /ABC
#END
```

In this example, when "XYZ <CR>" is received, "ABC" is sent.

Wait for the reception of a certain number of bytes of data

To send specific data when a certain number of bytes are received, write as follows.

```
#LOOP EVER
  #WAIT BYTE 20
  /ABC
#END
```

In this example, "ABC" is sent every time 20 bytes are received.

Multiple processes

Using processes can create complex conditions.

```
#PROCESS
  #LOOP EVER
    #WAIT TIME 1S
    /ABC
  #END
#PROCESS
  #LOOP EVER
    #WAIT BYTE 20
    /XYZ
  #END
```

In this example, "ABC" is sent every 1 second, and "XYZ" is sent every time 20 bytes are received.

```
#PROCESS
  #LOOP EVER
    #WAIT DATA /A
    /B
  #END
#PROCESS
  #LOOP EVER
    #WAIT DATA /X
    /Y
  #END
```

In this example, "B" is sent when "A" is received, and "Y" is sent when "X" is received.