Manual

QAM constellation configuration

Product Version v24.1

December 18, 2023

Imprint

PROCITEC GmbH
Rastatter Straße 41
D-75179 Pforzheim
Germany

Phone: +49 7231 15561 0
Fax: +49 7231 15561 11
Email: service@procitec.com
Web: www.procitec.com

Authorised Representative:
Dipl.-Ing. (FH) Dipl.-Inf. (FH) Jens Heyen

Registration Court:
HRB 504702 Amtsgericht Mannheim
Tax ID:
DE 203 881 534

Document ID:
PROCITEC-DSC-ConstellationXML_E-7f142bd669

Subject to technical modifications.

Contents

1. Introduction

In addition to the predefined constellations, the QAMn demodulator is able to demodulate arbitrary constellations and symbol-to-bit mappings defined by the user. The definition of the constellations is specified via XML, which is described in more detail in this document. The XML is part of the modem definition (*.ver file) for use in the APC.

2. Basic structure of XML

```
<?xml version="1.0"?>
<contree name="..." version="1.0">
    <constel name="...">
        <points>
            ...
        </points>
        <sym2bits>
            <sym2bit name="...">
                <absolute>
                    ...
                </absolute>
                <subset>
                    ...
                </subset>
                <differential>
                    ...
                </differential>
            </sym2bit>
        </sym2bits>
    </constel>
</contree>
```

Listing 2.1: Basic XML structure

The root element is <contree> with the attributes "name" and "version". Name can be chosen arbitrarily and is optional. The attribute "version" is used to specify the version of the constellation description format. Currently this must be 1.0. Another top-level XML tag within <contree> is <constel>, also with an arbitrary name which is used to identify the constellation within APC results. This second top-level tag is necessary due to compatibility reasons with other software and future extensions of the XML format.

The complex constellation points (IQ) are specified within <points>. Demapping of symbols to bits is possible absolutely, differentially and mixed. Differentially means that the resulting bits are dependent from two received consecutive symbols. This is specified within <sym2bits>.

3. Constellation points

The constellation symbols or points are defined within the XML-Tag <points>. The coordinates are entered one after the other separated by spaces. The real and imaginary parts of the complex number (IQ) are specified alternately. It's possible to use floating point and integer values. As delimiter character between the values newline and spaces or tabs should be used. As an example the definition of a QPSK constellation is given as follows:

```xml
<points>
    0.7071 0.7071  -0.7071 0.7071  -0.7071 -0.7071  0.7071 -0.7071
</points>

<!-- it is also possible to use multiple lines  -->
<points>
    0.7071  0.7071
   -0.7071  0.7071
   -0.7071 -0.7071
    0.7071 -0.7071
</points>
```

Listing 3.1: QPSK Constellation

4. Symbol-To-Bit demapping

Bit mapping is specified within the <sym2bits><sym2bit name="..."> tags. Again, this two level structure is necessary due to compatibility reasons <sym2bit>. Each mapping has to contain the elements <absolute>, <subset> and <differential> which describes the whole symbol to bit demapping. The absolute bit demapping is given within <absolute>. For each point defined in <points> a demapping entry is needed.

---

bit ordering
The order of the bits within the bit groups is most significant bit first (MSB first). This means that the right-hand digit of the bit group is the one occurring first in the data stream, both on the transmitting and receiving side. Therefore the resulting bitstream in the decoder (LSB first) appears mirrored per bit group or symbol.

For an absolute QPSK constellation the following would be appropriate:

```
<absolute>
    00 01 11 10
</absolute>
```

The order of bit groups aligns with the order in <points>. In above example 00 would be the bitpattern of point 1, 01 of point 2 and so on. The number of bits per symbol has to be fixed, but can be chosen arbitrary. If only parts of the bits should be demapped absolute the placeholder character "." has to be used in this position:

```
<absolute>
    0. 0. 1. 1.
</absolute>
```

Here, the second bit is determined in a differential fashion. This means, the bit demapping depends on the two last received symbols and not only on the last. If all bits should be demapped differential "." has to be used in all bit positions in the <absolute> tag:

```
<absolute>
    .. .. .. ..
</absolute>
```

The differential part is specified via the <subset> and <differential> elements. In <subset> each defined point has to be mapped to a subset. Multiple points can be mapped to the same subset. The numbering of the subsets must start with 0 and counts consecutively up to the number of subsets. In <differential> a matrix has to be given with the size "number of subsets" x "number of subsets". This describes the transition between 2 symbols or more precise between 2 subsets. The rows specifies the "from subset" and the columns the "to subset". Using the matrix, a bit mapping is given for each combination of 2 consecutive received subsets/symbols.

```
<subset>
    1 2 0 3
</subset>
<differential>
  <!--                 to subset      -->
  <!--                 0   1  2  3     -->

  <!--        0 --> 11 10 00 01
  <!-- from   1 --> 01 11 10 00
  <!-- subset 2 --> 00 01 11 10
  <!--        3 --> 10 00 01 11
</differential>
```

For a constellation with only absolute bit demapping a structure like the following is appropriate:

```
<sym2bits>
    <sym2bit name="abs_only">
        <!-- absolute bitmapping -->
        <absolute>
            00 10 11 01
        </absolute>
        <!-- all symbols in subset 0 -->
        <subset>
            0 0 0 0
        </subset>
        <!-- no differential parts, "." as placeholder
            1x1 matrix (1 subset only)
        -->
        <differential>
            ..
        </differential>
    </sym2bit>
</sym2bits>
```

Listing 4.1: Example for absolute mapping

For a constellation with only differential bit demapping a structure like the following is appropriate:

```
<sym2bits>
    <sym2bit name="diff_only">
        <!-- no absolute parts, "." as placeholder -->
        <absolute>
            .. .. .. ..
        </absolute>
        <!-- each symbol in different subset -->
        <subset>
            0 1 2 3
        </subset>
        <!-- 4 subsets, therefore 4x4 matrix
             row maps "from subset", column "to subset"
        -->
        <differential>
            00 01 11 10
            10 00 01 11
            11 10 00 01
            01 11 10 00
        </differential>
    </sym2bit>
</sym2bits>
```

Listing 4.2: Example for differential mapping

The whole <sym2bits> structure is optional and an absolute bit demapping is assumed when missing, where the bits are defined by the symbol order in the <points> tag. This could be useful for testing the demodulator or constellation only without caring about the result bits or when the demapping should be done in a following decoding step. So the first symbol specified in <points> is demapped to 000, the second to 001, 010, 011 and so on. The minimal number of bits is choosen to represent all symbols. As already mentioned, the order of the bits within the bit groups is MSB first. Therefore the resulting bitstream in the decoder (LSB first) appears mirrored per bit group or symbol.

5. Examples

5.1. Absolute bit demapping

```
<?xml version="1.0"?>
<contree name="abs" version="1.0">
    <constel name="abs">
        <points>
            0.7071  0.7071
           -0.7071  0.7071
           -0.7071 -0.7071
            0.7071 -0.7071
        </points>
        <sym2bits>
            <sym2bit name="abs">
                <absolute>
                    00 01 11 10
                </absolute>
                <subset>
                    0 0 0 0
                </subset>
                <differential>
                    ..
                </differential>
            </sym2bit>
        </sym2bits>
    </constel>
</contree>
```

Listing 5.1: Absolute bit demapping

received symbols	0.7071+j0.7071	-0.7071-j0.7071	0.7071-j0.7071
demapped bitsequence	00	11	10
resulting bitstream in decoder	001101 (LSB first)		

Table 1: Example of a received sequence

5.2. Differential bit demapping

```
<?xml version="1.0"?>
<contree name="diff" version="1.0">
    <constel name="diff">
        <points>
             1  0
             0  1
            -1  0
```

```
             0 -1
        </points>
        <sym2bits>
            <sym2bit name="diff">
                <absolute>
                    .. .. .. ..
                </absolute>
                <!--- note: custom order  -->
                <subset>
                    3 0 1 2
                </subset>
                <differential>
                    00 01 11 10
                    10 00 01 11
                    11 10 00 01
                    01 11 10 00
                </differential>
            </sym2bit>
        </sym2bits>
    </constel>
</contree>
```

Listing 5.2: Differential bit demapping

received symbols	1	-1	-j	-j	1
subsets	3	1	2	2	3
transisitions (initial is 0)	0 → 3	3 → 1	1 → 2	2 → 2	2 → 3
demapped bitsequence	10	11	01	00	01
resulting bitstream in decoder	0111100010 (LSB first)				

Table 2: Example of a received sequence

5.3. ASK2PSK8 or QAM8 with 2 amplitudes and 8 phases

```
<?xml version="1.0"?>
<contree name="ASKPSK" version="1.0">
    <!--
             *


         *       *

   *                   *

         *       *


             *
    -->
    <constel name="ASK2PSK8">
        <points>
             0.3333   0.3333
```

```
                 1.0000   0.0000
                 0.0000   1.0000
                -0.3333   0.3333

                 0.0000  -1.0000
                 0.3333  -0.3333
                -0.3333  -0.3333
                -1.0000   0.0000
            </points>
            <!--- no symbol to bit mapping here, mapping is natural bit order -->
        </constel>
</contree>
```

received symbols	0.33+j0.33	-j	j	-1
position in given list	0	4	2	7
demapped bitsequence	000	100	010	111
resulting bitstream in decoder	000001010111 (LSB first)			

Table 3: Example of a received sequence

5.4. Rectangular QAM16 with differential part as defined in V.22

```
<?xml version="1.0"?>
<contree name="QAM" version="1.0">
  <constel name="QAM16 V.22">
    <points>
       0.333333  0.333333
      –0.333333  0.333333
      –0.333333 –0.333333
       0.333333 –0.333333

       1.000000  1.000000
       0.333333  1.000000
      –0.333333  1.000000
      –1.000000  1.000000

      –1.000000  0.333333
      –1.000000 –0.333333
      –1.000000 –1.000000
      –0.333333 –1.000000

       0.333333 –1.000000
       1.000000 –1.000000
       1.000000 –0.333333
       1.000000  0.333333
    </points>
    <sym2bits>
      <sym2bit name="diff">
        <absolute>
          ..00 ..00 ..00 ..00 ..11 ..10 ..01 ..11
          ..10 ..01 ..11 ..10 ..01 ..11 ..10 ..01
        </absolute>
        <subset>
          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
        </subset>
        <differential>
          01.. 00.. 10.. 11.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01..
          11.. 01.. 00.. 10.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11..
          10.. 11.. 01.. 00.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10..
          00.. 10.. 11.. 01.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00..
          01.. 00.. 10.. 11.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01..
          01.. 00.. 10.. 11.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01..
          11.. 01.. 00.. 10.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11..
          11.. 01.. 00.. 10.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11..
          11.. 01.. 00.. 10.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11..
          10.. 11.. 01.. 00.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10..
          10.. 11.. 01.. 00.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10..
          10.. 11.. 01.. 00.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00.. 00.. 00.. 10..
          00.. 10.. 11.. 01.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00..
          00.. 10.. 11.. 01.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00..
          00.. 10.. 11.. 01.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01.. 01.. 01.. 00..
          01.. 00.. 10.. 11.. 01.. 01.. 00.. 00.. 00.. 10.. 10.. 10.. 11.. 11.. 11.. 01..
        </differential>
      </sym2bit>
    </sym2bits>
  </constel>
</contree>
```

received symbols	1 + j	0.333 - j	0.333 + j	-0.333 -0.333j
position in given list	4	12	5	2
subset	4	12	5	2
transitions (0 is initial)	0 → 4	4 → 12	12 → 5	5 → 2
absolute demapping	. . 11	. . 01	. . 10	. . 00
differential demapping	01 . .	11 . .	00 . .	10 . .
resulting bitsequence	0111	1101	0010	1000
resulting bitstream in decoder	1110101101000001 (LSB first)			

Table 4: Example of a received sequence

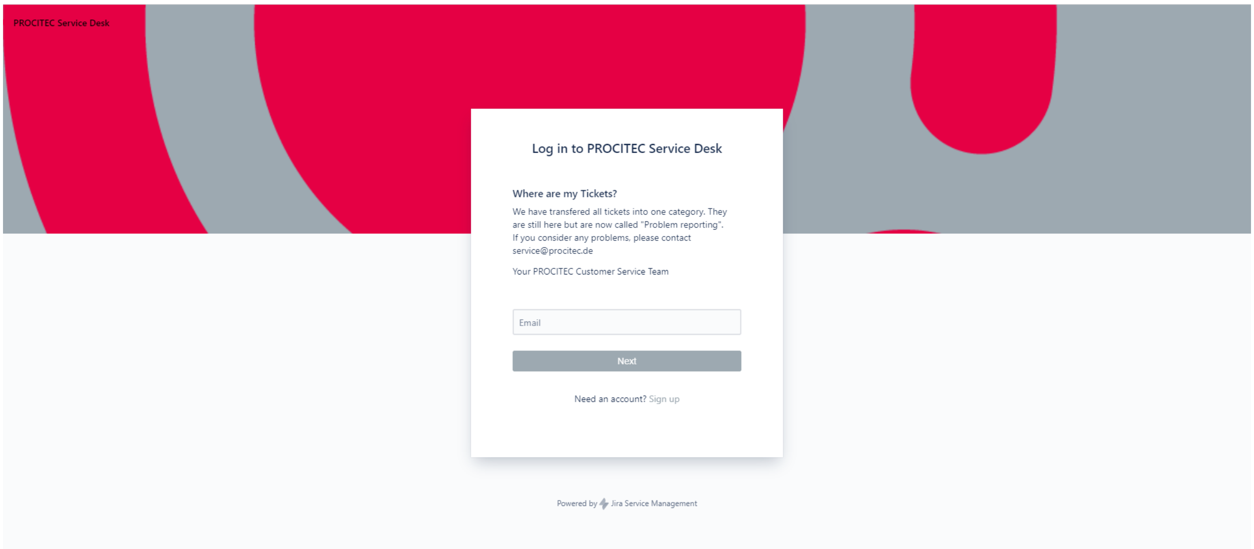
A. Support

Requests and suggestions?

All requests or suggestions regarding our go2signals product-range are very much appreciated; we would be delighted to hear from you.

Any questions? We are happy to assist you!

If you have any further questions, please do not hesitate to contact our Support Team for rapid assistance – just raise a service request at: **http://servicedesk.procitec.com**.



Log in to PROCITEC Service Desk

Where are my Tickets?

We have transfered all tickets into one category. They are still here but are now called "Problem reporting". If you consider any problems, please contact service@procitec.de

Your PROCITEC Customer Service Team

PROCITEC GmbH
Rastatter Straße 41
D-75179 Pforzheim
Phone: +49 7231 15561 0
Web: www.procitec.com
Email: service@procitec.com

List of Figures

List of Tables