1 GENERAL PRINCIPLE

The Mythic begins its frame with the headline :

MYTHIC X ;Y ;Z(;eventual parameters)[CR]

Where :
X is the Mythic Number (Maximum 2 characters).
Y is the user's identifiant login) (Maximum 10 characters).
Z is the frame identifiant.
The decimal separator is the dot (.)
The field separator is the dot comma (;°
The separator for the lines and the indicator for the end of frame is CR
The unbroken frames are:Les trames insécables sont :

- Request to send a result with result sent.
- Sent pack of calibration with results of calibration.

All the lines for a frame must be sent (if no information, the parameter is not filled).
All the identifiants (frames or parameters) are in Caps characters.
All the editable text field (Identification, lot) are coded in UTF8 (monitoring of the non latin languages).

<u>Maximum size of the fields which are used in the different frames.</u>

OPERATOR:	10 characters
USER:	10 characters
LOT (CALI or QC):	8 characters
SID:	16 characters
PID:	16 characters
ID:	20 characters

2 FRAMES

2.1.1 REQUEST TO SEND A RESULT

The Mythic send results after acknowledge of the host. This management enables to avoid a flow while the host is not connected in one hand and the insurance that it is capable to receive the size of the result on the other hand. Frames used in the "handshake" mode.

2.1.1.1 Request to send a result

MYTHIC X ;Y ;RESULT_READY ;Size[CR]

2.1.1.2 Acknowledge request to send results

ACK_RESULT_READY[CR]

2.1.1.3 Acknowledge result

ACK_RESULT;A;B[CR]
Where A : OK or Error Code, B : Reserved.

2.1.2 Routine result

MYTHIC X ;Y ;RESULT [CR]
DATE;30/10/2003 [CR]
TIME;15:36:38 [CR]
MODE;NORMAL [CR]
UNIT ; Unit [CR]
SEQ;352; SVM key [CR]
SID;3 [CR]
PID;X28 [CR]
ID;DUPONT [CR]
TYPE;STANDARD [CR]
TEST;DIF or CBC [CR]
OPERATOR ; login Mythic when analysis is done [CR]
PREL;OT [CR] (Sampling mode for M22; By default the value will be OT)
CYCLE;N [CR] (Cycle option. Possible values for M22 : N,A)
WBC; 11.0;A;B; 2.0; 4.0; 11.0; 15.0 [CR]] (Parameter; A; B; Low Panic ;Low ; High ;High Panic)
RBC; 6.00;A;B; 2.50; 4.00; 6.20; 7.00 [CR]
HGB;15.0;A;B; 8.5;11.0;18.8;20.0 [CR]
HCT;49.0;A;B;25.0;35.0;55.0;60.0 [CR]
MCV; 81.7;A;B; 70.0; 80.0;100.0;120.0 [CR]
MCH; 25.0;A;B; 25.0; 26.0; 34.0; 35.0 [CR]
MCHC; 30.6;A;B; 28.0; 31.0; 35.0; 36.0 [CR]
RDW; 8.1;A;B; 7.0; 10.0; 20.0; 25.0 [CR]
PLT; 320;A;B; 70; 150; 400; 500 [CR]
MPV; 7.5;A;B; 5.0; 6.0; 10.0; 12.5 [CR]
PCT; 0.4;A;B; 0.1; 0.2; 0.5; 0.6 [CR]
PDW; 9.9;A;B; 5.0; 8.0; 18.0; 25.0 [CR]
LYM%;30.5;A;B;15.0;25.0;50.0;55.0 [CR]
MON%;13.0;A;B; 1.0; 2.0;10.0;12.0 [CR]
NEU%;52.4;A;B;45.0;50.0;80.0;85.0 [CR]
EOS%;2.4;A;B; 0.0;0.5;1.0; 5.0 [CR]
BAS%;1.6;A;B; 0.0;0.5;1.0; 5.0 [CR]
LYM; 3.4;A;B; 0.8; 1.0; 5.0; 5.5 [CR]
MON; 1.4;A;B; 0.0; 0.1; 1.0; 1.1 [CR]
NEU;6.4;A;B;4.0;5.0;8.0;9.0 [CR]
EOS;1.4;A;B; 0.0;0.5;1.0; 5.0 [CR]
BAS;0.6;A;B; 0.0;0.5;1.0; 5.0 [CR]
WBC CURVE; value channel1;… ;value channel 128 [CR]
WBC THRESHOLDS;S1;S2;S3 [CR] (S3 in not present in Mythic 22)
RBC CURVE; value channel1;… ;value channel 128 [CR]
RBC THRESHOLDS; S1;S2 [CR]
PLT CURVE; value channel1;… ;value channel 128 [CR]
PLT THRESHOLDS; S1 [CR]
ALARMS;x;x;x;x;x;x;x;x;x;x;x [CR]

INTERPRETIVE_WBC;y;y;y;y;y;y;y;y;y;y;y;y [CR]
INTERPRETIVE_RBC;y;y;y;y;y;y;y;y;y;y;y;y [CR]
INTERPRETIVE_PLT;y;y;y;y;y;y;y;y;y;y;y;y [CR]
COMMENT; message RUO if necessary [CR]
LMNE MATRIX; [CR]
m;m;m;m;m;m;m;m;m;m;m;m;m m;m;m;m;m; [CR]
m;m;m;m;m;m;m;m;m;m;m;m;m;m;m;m;m;m;m;m;m m;m;m;m;m; [CR]
…………
m ;m ;m ;m ;m ;m ;m ;m ;m ;m ;m ;m ; [CR]
[CR]
LMNE SHADE MATRIX; [CR]
m;m;m;m;m;m;m;m;m;m;m;m;m m;m;m;m;m; [CR]
m;m;m;m;m;m;m;m;m;m;m m;m;m;m;m; [CR]
…………
m ;m ;m ;m ;m ;m ;m ;m ;m ;m ;m ;m ; [CR]
[CR]
THRES 5D LMNE MATRIX; [CR]
T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;[CR]
T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;T ;S ;[CR]
END_RESULT; value of the checksum [CR]

Alarms and flagging

A : None or *
B : None or D or H or h or l or L
x : None or Alarm
y : None or pathologic flag.

Sampling mode for Mythic 22 : PREL;p

Value for p	Instrument type	Comment
"OT"	M22 open tube	Sampling on "Open Tube"
"NO RUN"	M22 open tube and closed tube	Unknown, the result doesn't contain any sampling information
"OV"	M22 closed tube	Sampling "Other Vial"
"CT"	M22 closed tube	Sampling "Closed tube"

Cycle option for Mythic 22: CYCLE;o

Value for o	Instrument type	Comment
"N"	M22 open tube and closed tube	Standard cycle is used
"A"	M22 open tube and closed tube	Important altitude cycle is used

Curves and thresholds

S1 S2 S3 : thresholds of the curve (between 0 and 127). S3 is only present in Mythic 18 (LMG).
This frame is for a LMG test, the CBC and DIFF cases must be considered for thh Mythic 22.
The numerical values can be replaced by +++++ (5 plus) when the results is higher than the limit or by ….. (5 dots) when the result is invalid.
If the RUO are not considered, then the parameters PCT and PDW are not sent.
m : field ASCII from 1 to 4 characters from the compression of the datas of the scattergram. The lines have a variable size but no more than 250 characters. The number of lines is correlated to the rate of the compression. See after for a full description.
T : Named of a scattergram threshold. See detail hereafter.
S : Value for scattergram threshold. See detail hereafter.

Format for a line of haematological parameter

WBC; 11.0 ;A ;B ; 2.0 ; 4.0 ; 11.0 ; 15.0 [CR]

Order	Field designation	Possible values
1	Name of the parameter	WBC
2	Value for the parameter in specified unit	11.0 +++++ (5 plus) if higher than capacity. ….. (5 dots) if invalidated result
3	Flag A	None or "*" if rejected result
4	Flag B	None 'D' if result higher than capacity 'H' if result higher than panic value 'h' if result higher than normal value 'l' if result lower than normal value 'L' if result lower than panic value
5	Low panic value	2.0
6	Low normal value	4.0
7	High normal value	11.0
8	High panic value	15.0

2.2 List of flags

S-UP NOT DONE
S-UP FAIL
QC NOT DONE
QC FAIL
INS-M
INS-T
INS-P
INS-R
INS-H

L1
L2
L3
L4
L5
R1
R2
P1
P2
P3
NH
NL
RL
N1
N2
IC
HL
W-CL
R-CL
O-CL
OPT-

2.3 List of the WBC Pathologic flags

LEU>
LEU<
LYM>
LYM<
MON>
NEU>
NEU<
EOS>
BAS>
MYEL
LIC
ALY
ERYB
BLST
NO_INTERPRETATION

2.4 List of the RBC Pathologic flags

ANE
ERY>
MICRO

MACRO
MICR>
MICR>>
MICR>>>
MACR>
MACR>>
MACR>>>
ANIS>
ANIS>>
ANIS>>>
HYPOCR
COLDAGG
NO_INTERPRETATION

2.5 List of the PLT Pathologic flags

THR>
THR<
MACROP
PLTAGGR
MICROC
SCHIZ
CELLD
NO_INTERPRETATION

2.6 Scattergram details and thresholds

The transmitted scattergram is decomposed in 2 compressed scattergrams. This combinaison enables to code 2 possible levels for the 3rd dimension : Low dot or high.
After decompression (with the same algorithm), these scattergrams represent :

- LMNE MATRIX: representation 2D of the high dots, 1 bit per dot, 128 x 128 bits.
- LMNE SHADE MATRIX: representation 2D of the low dots, 1 bit per dot, 128 x 128 dots.

LMNE MATRIX, bit i	LMNE SHADE MATRIX, bit i	Points or particules
0	0	absence
0	1	Low level
1	0	Presence
1	1	Impossible case

Principle of the compression algorithm :

- The repetition of the byte 00 are coded « Zxxx », xxx = counter in ASCII Hexadecimal.
- The repetition of the byte FF are coded « Uxxx », xxx = counter in ASCII Hexadecimal.
- The other bytes are formatted in ASCII Hexadecimal : « xx »,.
- The end of the scattergram finshes with the character « T ».

- Each item is separed by a delimiter « ; ».
- Before to reach 250 characters; a return [CR] is inserted.

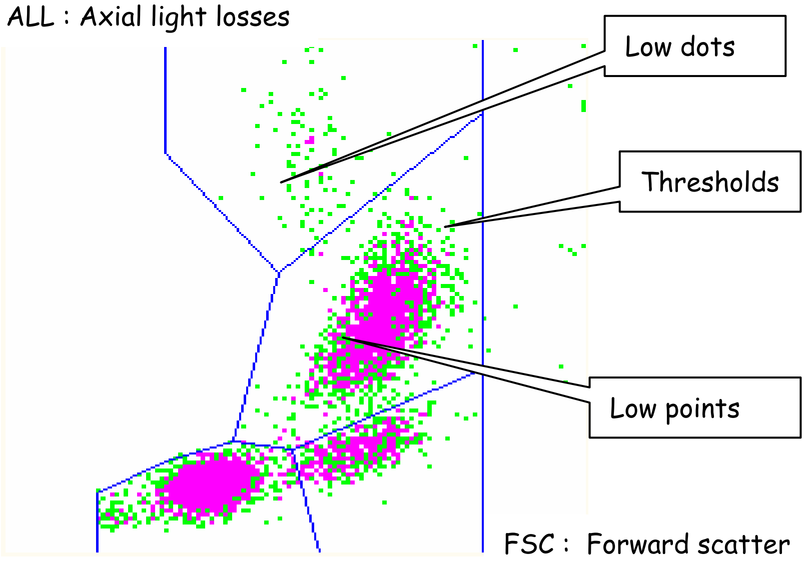
Example of compatible decompression function (Language C):

```
int UnSqueezeMatrix(unsigned char *pucDestMatrix, unsigned char *pucSrcM)  // dest, src
{
int iCnt;
unsigned char *pucDest, *pucSrc, ucVal;
pucSrc  = pucSrcM;    // source
pucDest = pucDestMatrix;   // dest
memset(pucDestMatrix, 0, sizeof(BMP_MATRIX));    // maz dest
while (*pucSrc != 'T')    // end of trame
   {
   ucVal = *pucSrc;
   switch(ucVal)
      {
      case '\x0d':  // ignore CR , LF , separator
      case '\x0a':
      case ';':
         break;
      case 'U':     // value 0xFF
         {
         pucSrc++;
         sscanf((char *)pucSrc,"%x", &iCnt);
         memset(pucDest, 0xFF, iCnt);
         pucDest+= iCnt;
         break;
         }
      case 'Z':     // value 0
         {
         pucSrc++;
         sscanf((char *)pucSrc,"%x", &iCnt);
         memset(pucDest, 0x00, iCnt);
         pucDest+= iCnt;
         break;
         }
      default:
         {
         sscanf((char *)pucSrc,"%x", &iCnt);
         *pucDest++ = (u_char)iCnt;
         break;
         }
      }
   while (*pucSrc != ';')        // skip ; separator
      pucSrc++;
```

```
  pucSrc++;
  }
return((int)pucDest - (int)pucDestMatrix);
}
```

This function needs that « ; » is considered as a delimiter by the function scanf(), which is normally done in most of the systems.

Graphic representation for the scattergram:



Considering a blank bitmap map with an origin (0,0) in the left upper part, a basic layout of the scattergram can be done by the following code <u>(Language C):</u>

```
static u_char ucMask[] = {0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
UnSqueezeMatrix( pucMat,  bufferM1);    // dest, src  uncompress
UnSqueezeMatrix( pucMatL, bufferM2);    // dest, src  uncompress
for (i = 0; i <(16*128); i++)       // 128 x 128 pixels
  {
  ucVal  = *(pucMat+i);        // LMNE MATRIX  byte
  ucValL = *(pucMatL+i);       // LMNE SHADE MATRIX byte
  iX = (i%16)*8;
  iY = i/16;
  for (k=0; k<8; k++)    // MSB to LSB
    {
    if ((ucMask[k] & ucValL) != 0)    // low level
      img->Canvas->Pixels[iX+k]  [iY]   = ColorLow;
    else if ((ucMask[k] & ucVal) != 0)    // high level
      img->Canvas->Pixels[ iX+k]  [iY]    = ColorHigh;
    }
  }
```

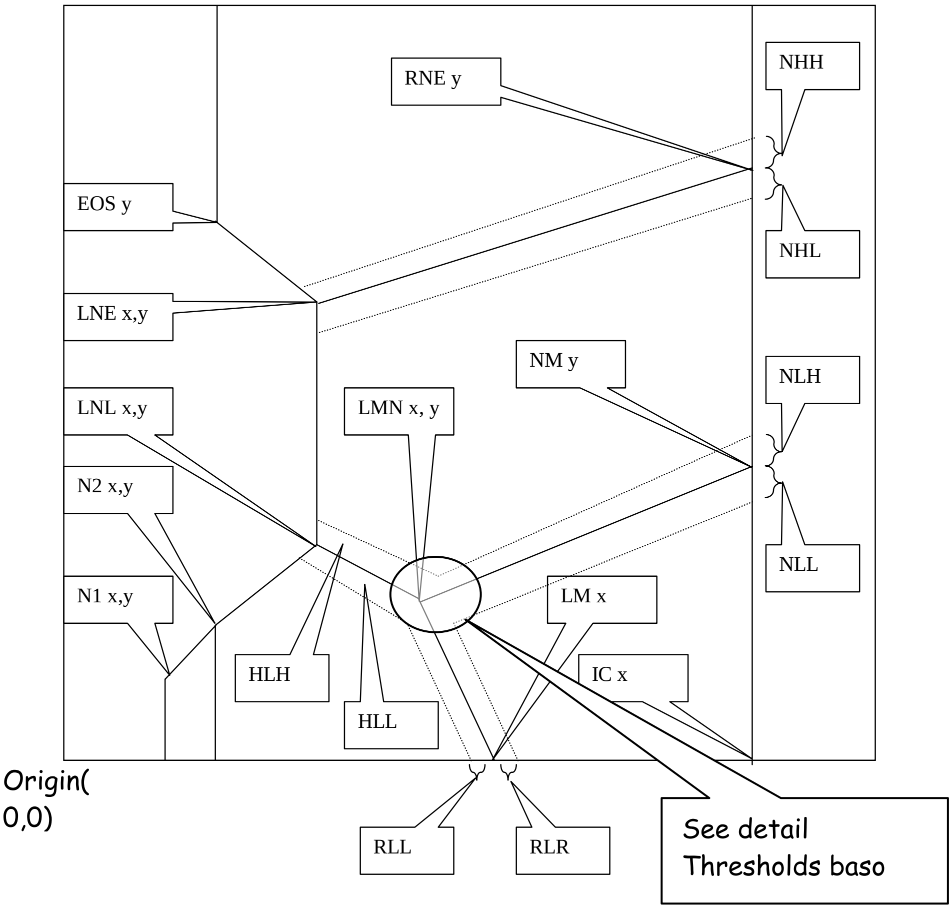
Note that the point (0,0) of the graphic plot is accesssible in [127][0] from the decompressed scattergrams (128 bytes ALL , 8 bytes FSC) .

Thresholds of the scattergram:
They are on 2 lines, they are Ils sont sur 2 lignes, ils sont preceded by their wordings and represent their co-ordinates X,Y on the graphic plot of the scattergram : origin in the left lower part.

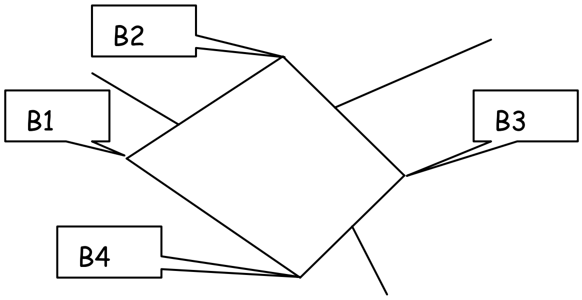
Title	Description	Range mini / Maxi	Value by default
N1X	Threshold N1X	0 / 127	20
N1Y	Threshold N1Y	0 / 127	15
N2X	Threshold N2X	0 / 127	35
N2Y	Threshold N2Y	0 / 127	23
LnlX	Threshold LNLX	0 / 127	50
LnlY	Threshold LNLY	0 / 127	28
LmnX	Threshold LMNX	0 / 127	63
LmnY	Threshold LMNY	0 / 127	26
LneX	Threshold LNEX	0 / 127	55
LneY	Threshold LNEY	0 / 127	60
EosY	Threshold EOSY	0 / 127	90
LmX	Threshold LMX	0 / 127	69
ICX	Threshold ICX	0 / 127	105
NmY	Threshold NMY	0 / 127	50
RneY	Threshold RNEY	0 / 127	110
BasoX1	Thresholds B1X	0 / 127	50
BasoY1	Threshold B1Y	0 / 127	25
BasoX2	Threshold B2X	0 / 127	52
BasoY2	Threshold B2Y	0 / 127	33
BasoX3	Threshold B3X	0 / 127	68
BasoY3	Threshold B3Y	0 / 127	29
BasoX4	Threshold B4X	0 / 127	65
BasoY4	Threshold B4Y	0 / 127	22
NHL	Threshold NHH	0 / 127	2
NHH	Threshold NHH	0 / 127	2

NLL	Threshold NLL	0 / 127	2
NLH	Threshold NLH	0 / 127	2
RLL	Threshold RLL	0 / 127	2
RLR	Threshold RLR	0 / 127	3
HLL	Threshold HLL	0 / 127	2
HLH	Threshold HLH	0 / 127	2

Position of the Thresholds (except baso) on the scattergram:



Position of the Thresholds baso on the scattergram



Extract of the scattergram frames and thresholds (example):

```
…………..
LMNE MATRIX;
Z105;10;Z70;4;Zf;8;Z2f;2;Z2f;20;Z2;40;………..;1;0;8;1;80;Zc;8;Ze;2;2;84;Zc;20;0;40;Zf;4;88;Zd;1;20;Zd;
44;Ze;1;18;Z3;20;Z9;1;1;Z2;10;Zc;1;50;Zd;……………………………………………………………………;10;5;Zf;24;20;2;Z3;
Z2;36;70;92;Za;4;Z2;4;a8;6;20;Zc;………………9;40;Zd;a;25;e4;Zb;4;0;25;fe;Zc;4;1;c0;b9;80;Zc;8;99;84;Za;
e;66;38;a0;Zb;1;1a;6d;1c;2;Zb;11;18;…………………;46;36;c4;10;Za;8;9;24;b3;c0;2;Zb;4;a3;e7;5;Z2;2;Z9;
Z3;c;5d;cc;Z8;1;82;40;Z2;84;60;e8;Z8;……………..;a4;Z2;10;95;28;4c;Z8;7;8b;33;30;1;0;76;4;Z9;65;0;6c;2;
c0;2b;5a;30;Zc;d1;60;Zd;1;12;8;Ze;92;20;Ze;5b;Z1f;20;Z8c;T;
LMNE SHADE MATRIX;
Z105;10;Z70;4;Z72;40;Zc;10;2;……………………………………………………………………………………;80;Zb;2;Z10;80;
;0;80;Za;20;Z4;80;Zb;80;Z3;80;Z9;……….;10;Z14;20;2;Zb;20;2;62;50;Zd;2;Zc;4;Z2;4;88;4;Zd;8;2e;
40;23;60;Zc;1;44;14;90;Zb;1;8;………..;1;12;44;8;2;Zb;1;18;92;10;2;Za;10;0;42;20;4;10;Zd;b1;Ze;a2;2;4;Z2;2;
92;Zf;19;Z1f;20;Z8c;T;
THRES 5D LMNE MATRIX;
N1X;20;N1Y;15;N2X;35;N2Y;23;EosY;90;LneX;55;LneY;60;LnlX;50;LnlY;28;RneY;110;NmY;46;LmX;69;LmnX;63;LmnY;26;ICX;105;
BasoX1;60;BasoY1;26;BasoX2;64;BasoY2;30;BasoX3;70;BasoY3;29;BasoX4;65;BasoY4;20;NHH;2;NHL;2;RLL;2;RLR;3;NLH;2;NLL;2;HLH;
2;HLL;2;
……………..
```

2.7 CALIBRATION

MYTHIC X;Y;CALIBRATION;Y ;Date of calibration ;time of calibration;LOT;Expiry date; Creation date ;Creation time ;X ;A ;B ;C ;D ;F ;Number of results [CR]
WBC; target ; limit [CR]
RBC; target ; limit [CR]
HGB; target ; limit [CR]
HCT; target ; limit [CR]
PLT; target ; limit [CR]
END_CALI ; checksum [CR]
Where
X : User who creates the lot.
Y : User who calibrates.
A, B, C, D and F are the calibration coefficients for, WBC, RBC, HGB, HCT et PLT.
Follow with the N results which served for calibration.
When the coefficients are manually input, the informations for the lot are blank and the number of results is null.
The acknowledge for the reception of the calibration is done by the frame
ACK_CALI ;Lot;A [CR]
Where A : OK or Error code.

2.7.1 Results

MYTHIC X ;Y ;RESULT [CR]
DATE;30/10/2003 [CR]
TIME;15:36:38 [CR]

MODE;CALIBRATION [CR]
UNIT ; Unité [CR]
SEQ;352; 0 [CR]
TEST;DIF [CR]
OPERATOR ; login Mythic when the analysis is done [CR]
WBC; 11.0; A;B;;;; [CR]
RBC; 6.00; A;B;;;; [CR]
HGB;15.0; A;B;;;; [CR]
HCT;49.0; A;B;;;; [CR]
PLT; 320; A;B;;;; [CR]
END_RESULT;Checksum value [CR]

A and B same as routine result.

2.8 QC

2.8.1 Result

MYTHIC X ;Y ;RESULT [CR]
DATE;30/10/2003 [CR]
TIME;15:36:38 [CR]
MODE;QC [CR]
UNIT ; Unit [CR]
SEQ;352; 0 [CR]
LOT, Lot number [CR] (Maximum 10 characters)
LOT DATE; Creation Date for the lot ; Creation time [CR]
EXPIRY DATE; Expiry date [CR]
USER; User (login) who creates the lot [CR]
TEST;DIF or CBC [CR]
OPERATOR ; login Mythic when the analysis is done [CR]
WBC; 11.0;A;B; 2.0;;;15.0 [CR]
RBC; 6.00;A;B; 2.50;;; 7.00 [CR]
HGB;15.0;A;B; 8.5;;;20.0 [CR]
HCT;49.0;A;B;25.0;;;60.0 [CR]
MCV; 81.7;A;B; 70.0;;;120.0 [CR]
MCH; 25.0;A;B; 25.0;;;35.0 [CR]
MCHC; 30.6;A;B; 28.0;;;36.0 [CR]
RDW; 8.1;A;B; 7.0;;;25.0 [CR]
PLT; 320;A;B; 70;;;500 [CR]
MPV; 7.5;A;B; 5.0;;;12.5 [CR]
THT; 0.4;A;B; 0.1;;;0.6 [CR]
PDW; 9.9;A;B; 5.0;;;25.0 [CR]

```
LYM%;30.5;A;B;15.0;;;55.0 [CR]
MON%;13.0;A;B; 1.0;;;12.0 [CR]
NEU%;52.4;A;B;45.0;;;85.0 [CR]
EOS%;2.4;A;B; 0.0;;; 5.0 [CR]
BAS%;1.6;A;B; 0.0;;;5.0 [CR]
LYM;  3.4;A;B;  0.8;;;5.5 [CR]
MON;  1.4;A;B; 0.0;;;1.1 [CR]
NEU;6.4;A;B;4.0;;;9.0 [CR]
EOS;1.4;A;B; 0.0;;;5.0 [CR]
BAS;0.6;A;B; 0.0;;;5.0 [CR]
END_RESULT; checksum value [CR]
```

2.9 REPEATABILITY

2.9.1 Result

```
MYTHIC X ;Y ;RESULT [CR]
DATE;30/10/2003 [CR]
TIME;15:36:38 [CR]
MODE;REPEATABILITY [CR]
UNIT ; Unit [CR]
SEQ;352; 0 [CR]
TEST;DIF [CR]
OPERATOR ; login Mythic when the analysis is done [CR]
WBC; 11.0;A;B;;;; [CR]
RBC; 6.00; A;B;;;; [CR]
HGB;15.0; A;B;;;; [CR]
HCT;49.0; A;B;;;; [CR]
MCV; 81.7; A;B;;;; [CR]
MCH; 25.0; A;B;;;; [CR]
MCHC; 30.6; A;B;;;; [CR]
RDW;  8.1; A;B;;;; [CR]
PLT;  320; A;B;;;; [CR]
MPV;  7.5; A;B;;;; [CR]
THT;  0.4; A;B;;;; [CR]
PDW;  9.9; A;B;;;; [CR]
LYM%;30.5; A;B;;;; [CR]
MON%;13.0; A;B;;;; [CR]
NEU%;56.4; A;B;;;; [CR]
EOS%;56.4; A;B;;;; [CR]
BAS%;56.4; A;B;;;; [CR]
LYM;  3.4; A;B;;;; [CR]
MON;  1.4; A;B;;;; [CR]
```

NEU; 6.2; A;B;;;; [CR]
EOS; 6.2; A;B;;;; [CR]
BAS; 6.2; A;B;;;; [CR]
END_RESULT; Checksum value [CR]

3 CHECKSUM

The schecksum is a Cyclic Redundancy Check (CRC-16)

3.1 Algorithm

The used code is reported as below (code in C).
It allows the calculation of the generation of CRC-16 standard.

This alogorith is mainly used for embedded system (like disk controlor)
For better performances, the calculation is done with a research table of 16 values.
We obtain for the CRC calculation :

```
CRC = 0xFFFF
For each byte
      Work on quartet Most Significant Bit (MSB):
      Indice = byte XOR CRC
      Indice = Indice AND 000F
      CRC = Table(Indice) XOR (CRC divided by 16)
      Work on quartet Least Significant Bit (LSB):
      Indice = byte divided by 16
      Indice = Indice XOR CRC
      Indice = Indice AND 000F
      CRC = Table(Indice) XOR (CRC divided by 16)
```

<u>Example of implementation in language C :</u>
Declaration of the research table :

```
static const unsigned short ausCrcTab1[] =
 {
 0x0000, 0xCC01,0xD801,0x1400,0xF001,0x3C00,0x2800,0xE401,
 0xA001,0x6C00,0x7800,0xB401,0x5000,0x9C01,0x8801,0x4400,
};
```

Calculation of CRC.

```
unsigned short calc_crc(unsigned char *pucData, long lSize)
{
unsigned short usAcc1 = 0xFFFF;

 while ( lSize > 0 )
      {
```

```
      /* gestion par quartet du calcul  */
      usAcc1 = ausCrcTab1[(*pucData ^ usAcc1) & 15] ^ (usAcc1 >> 4);
      usAcc1 = ausCrcTab1[((*pucData >> 4) ^ usAcc1) & 15] ^ (usAcc1 >> 4);

      pucData++;
      lSize--;
      }

return(usAcc1);
}
```

^ : XOR.
>> : Logic shift to the right.
& : And logic.

The CRC is calculated from the begining of the datas to the last line ([CR] included) preceding the checksum line (title + value).