Application Note 005e

AED / FIT®

Measurment query via the serial link (RS232/RS485)

I1699-1.1 en

HBM

Content

1 Introduction

The FIT® digital load cell and the AED transducer electronics (AD103) work internally at very high sampling rates (up to 1200 measured values / second). Integrating these electronics in an overall system always raises questions such as the data transfer rates that can be implemented over the serial connection. This application document provides answers to these data output questions and is directed in particular at software engineers.

In the text below, the electronics are referred to as AED.

2 Data formats and output speed

Data output is requested with the command MSV?.

The AED basically supports three output formats which are set using the COF command:

- 2-byte binary output
- 4-byte binary output
- ASCII output

2-byte binary output is the fastest, but it has the disadvantage that it is not possible to transfer a measurement status. Resolution, relative to the measuring range, is max. ±30000 d plus the overflow reserve of 2765 d. For scales with 3000 d, this resolution is usually sufficient. The measuring range is defined between zero and the max. capacity of a scale (after the **LDW/LWT** adjustment, previous load already removed).

As well as the possible higher resolution, 4-byte binary output has the advantage of being able to transfer the measurement status (see **MSV** command description) (**COF**8).

Because of the number of characters that need to be transferred, ASCII output is not suited to fast data output.

The remarks below relate to 4-byte binary output for FIT® types and AD103.

The AED output rate for measurement data depends on the filter setting (**HSM**, **FMD**, **ASF** and **ICR**). For working at a high output rate, **ICR**0 should always be the setting. The commands FMD and ASF are then crucial to the output rate. Using the **ICR** command for mean-value calculation vastly improves the filter effect.

The following table shows the sampling rate of the AED in relation to **FMD** and **ASF** for **ICR**0 (valid for second generation FIT®, AD103, **HSM**0):

Filter setting	FMD0,2,3,4	FMD1
ASF0	600 M/s	600 M/s
ASF1	600 M/s	600 M/s
ASF2	600 M/s	300 M/s
ASF3	600 M/s	200 M/s
ASF4	600 M/s	150 M/s
ASF5	600 M/s	125 M/s
ASF6	600 M/s	100 M/s
ASF7	600 M/s	85 M/s
ASF8	600 M/s	75 M/s
ASF9	600 M/s	66 M/s

*) M/s = measurements/second

The measurement time [s] of the AED is = 1/sampling rate [M/s] *)

The measurement time is increased by the integration time (command **ICR**>0) (see the **MSV**? command description).

As well as the measurement time, the transmission speed of the interface is crucial for the measurement data output rate .

The transmission speed of the interface is the baud rate, **BDR**. This does not alter the number of measured values that the AED determines every second.

Baud rate	Transmission time for one byte/character
2400	4.4 ms
4800	2.2 ms
9600	1.1 ms
19200	0.6 ms
38400	0.3 ms

With this information, it is possible to estimate the transmission time for a command sequence. To do this, determine the number of characters in the command and multiply by the transmission time.

In addition to this, the AED has a processing time for each command (command interpretation). These times can be found in the individual command descriptions (total time = transmission time + processing time).

If the sampling rate is higher than the transmission rate, bits six and seven are set in the measurement status. This means that not all the measured values have been read out and thus the old measured value in the output buffer has been overwritten.

3 Measurement query without bus mode

If only one AED is connected to the Master, bus mode is not available. So communication can take place without the Select command.

There are basically two query modes:

- Query using individual commands MSV?;
- Query using block output of n measured values with **MSV**?n; (n=0...65000)

In addition to this, there is automatic data output (see automatic data output section).

Query using individual commands MSV?;

Example:

	Command	**Measurement time (with ICR0, HSM0)**	**Response (withCOF8)**
Master	MSV?;		
AED / FIT®		1.66 ms + 1.66 ms	xxxs crlf

xxx - measured value, s - status,
crlf – end label (0dh, 0ah)

The response time of the AED to a single measurement query MSV?; is:

Response time = transmission time for, MSV?;'
+ 1.6 ms (command interpretation)
+ measurement time
+ transmission time for the measured value (4 bytes + end label crlf)

The transmission time is dependent on the baud rate set, BDR. From this formula, it is obvious that with single measured value query, it is not possible to achieve the maximum sampling rate of 600 measured values/s, as the transmission times keep adding up.

Response time = 5 chars ∗ 0.3 ms + 1.66 ms + 1.66 ms + 6 chars ∗ 0.3 ms = 6.6 ms

This means that the maximum data transfer rate that can be achieved with single data query is <150 measured values/s (BDR38400, ICR0, measurement time = 1.6 ms).

Query using block output of n measured values with MSV?n;

Example:

(n=5)

	Command	Measurement time (for ICR0, HSM0)	Response (for COF8)
Master	MSV?5;		
AED / FIT®		1.66 ms + 1.66 ms 1.66 ms 1.66 ms 1.66 ms 1.66 ms	xxxs xxxs xxxs xxxs xxxs crlf

xxx - measured value, s -status,
crlf - end label (0dh, 0ah)

The end label crlf (0dh, 0ah) is only output with the last measured value. The time interval between the measured values is precisely the measurement time (1.66 ms). Only once the command has been received is the max. 1.66 ms of interpretation time available for the command.

The block response time of the AED to a measurement query MSV?n; (n=0…65000) is

Block response time = transmission time for ‚MSV?n;'

+ 1.6 ms (command interpretation)

+ n ∗ tx (tx = measurement time)

The output times for the measured values are not added to the measurement time, as output and measurement take place simultaneously (condition: high baud rate, transmission of 4 bytes < 1.66 ms).

With this measurement query in a block, it is possible to achieve the high data transfer rate of 600 M/s (BDR38400). Block output can be stopped at any time by means of the STP command (exception: RS485, 2-wire connection). If output of a measured data value has started it will be completed in full.

4 Measurement query in bus mode (RS485 interface)

If several AEDs are connected to the Master, bus mode is available. Communication requires the Select command to be used, to address the individual bus nodes.

There are basically two query modes:

- Data query using single **MSV**?; commands with time-synchronized measurement of all bus nodes
(**COF**0...**COF**12, measured values with end label crlf)

- Data query using single **MSV**?; commands with time-synchronized measurement of all bus nodes
(**COF**32...**COF**44, measured values without end label)

- Data query in bus mode (**COF**16...**COF**28) without time-synchronized measurement

Automatic data output must not be used.

4.1 Data query using single MSV?; commands with time-synchronized measurement (COF0...COF12)

Time-synchronized measurement is initiated by the command sequence **S**98;**MSV**?;. The 'broadcast' command, **S**98 followed by the **MSV**?; command, starts measurement immediately in all AEDs. The measured values are stored in the output memory and can then be read out by using the Select commands. For a high output rate, the setting should be **ICR**0. Taking four bus nodes as an example (AED) gives the following query pattern:

1. Master initiates bus mode: **S**98;**COF8**;**ICR**0;
2. Measured values read out in cycles

	Command	Measurement time (ICR0, HSM0)	Output 1. AED	Read out 2. AED	Read out 3. AED	Read out 4. AED
Master	S98;MSV?S01;			S02;	S03;	S04;
1. AED		1.6 ms+1.6 ms	xxxscrlf			
2. AED		1.6 ms+1.6 ms		xxxscrlf		
3. AED		1.6 ms+1.6 ms			xxxscrlf	
4. AED		1.6 ms+1.6 ms				xxxscrlf

The string '**S**98;**MSV**?;y01;' can be output in one piece. After the interpretation and measurement time, AED 1 responds with its measured value. Once this value is received, AED 2 can be read out. The response comes immediately after the Select command is received, as the measured value is already in the AED output buffer.

After the last AED has been read out, the cycle begins again, with '**S**98;**MSV**?;**S**01;' .

At a baud rate of 38400 bd and an output rate of **HSM**0,**ICR**0, **FMD**0 (any **ASF**), with four bus nodes, the cycle time is (see previous table):

Cycle time 4 > = 12 chars ∗ 0.3 ms + 3.3 ms + 6 chars ∗ 0.3 ms + 3 ∗ 10 chars ∗ 0.3 ms

'**S**98;**MSV**?;**S**01;'	Response AED 1	Read out AED 2...4
	(COF8)	
	xxxscrlf	3 x (**S**xx; yyyscrlf)

cycle time4 > = 17.7 ms

At a baud rate of 38400 bd and an output rate of **HSM**0,**ICR**0, **FMD**0 (any **ASF**), with eight bus nodes, the cycle time is:

Cycle time 8 > = 12 chars ∗ 0.3 ms + 3.3 ms + 6 chars ∗ 0.3 ms + 7 ∗ 10 chars ∗ 0.3 ms

'**S**98;**MSV**?;**S**01;'	Response AED 1	Read out AED 2...8
	(**COF**8)	
	xxxscrlf	7 x (**S**xx; yyyscrlf)

cycle time4 > = 29.7 ms

4.2 Data query using single MSV?; commands with time-synchronized measurement of all bus nodes (COF32...COF44)

With output mode **COF**32...44, the end label crlf is omitted when the measured values are output.

1. Master initiates bus mode: **S**98;**COF40**;**ICR**0;
2. Measured values read out in cycles

	Command	Measurement time (ICR0, HSM0)	Output 1. AED	Read out 2. AED	Read out 3. AED	Read out 4. AED
Master	**S**98;**MSV**?**S**01;			**S**02;	**S**03;	**S**04;
1. AED		1.6 ms+1.6 ms	xxxs			
2. AED		1.6 ms+1.6 ms		xxxs		
3. AED		1.6 ms+1.6 ms			xxxs	
4. AED		1.6 ms+1.6 ms				xxxs

The string ‘**S**98;**MSV**?;**S**01;’ can be output in one piece. After the interpretation and measurement time, AED 1 responds with its measured value. Once this value is received, AED 2 can be read out. The response comes immediately after the Select command is received, as the measured value is already in the AED output buffer.

After the last AED has been read out, the cycle begins again, with ‘**S**98;**MSV**?;**S**01;’ .

At a baud rate of 38400 bd and an output rate of **HSM**0, **ICR**0, **FMD**0 (any **ASF**), with four bus nodes the cycle time is:

Cycle time 4 > = 12 chars ∗ 0.3 ms + 3.3 ms + 4 chars ∗ 0.3 ms + 3 ∗ 8 chars ∗ 0.3 ms

‘**S**98;**MSV**?;**S**01;’	Response AED 1	Read out AED 2...4
	(**COF**40)	
	xxxs	3 x (**S**xx; yyys)

cycle time4 > = 15.3 ms

At a baud rate of 38400 bd and an output rate of **HSM**0,ICR0, FMD0 (any ASF), with eight bus nodes the cycle time is:

Cycle time 8 > = 12 chars ∗ 0.3 ms + 3.3 ms + 4 chars ∗ 0.3 ms + 7 ∗ 8 chars ∗ 0.3 ms

‘**S**98;**MSV**?;**S**01;’	Response AED 1	Read out AED2...8
	(**COF**40)	
	xxxs	7 x (**S**xx; yyys)

cycle time4 > = 25 ms

These times are regarded as the minimum times for the cycle and do not take into account any delay times that may occur in the Master (program runtimes).

With the setting **FMD**1, **ASF** > 0, the measurement time increases (see the data formats and output speed section).

4.3 Data query in bus mode (COF16...COF28) without time-synchronized measurement

If time synchronized measurement between AEDs is not required, bus mode can be used. Measurement is started once, by the command **MSV**?0;.

The AEDs then store the completed measured value in the output memory. If a measured value is not read out, it will be overwritten by the next measured value to be completed. The only way to stop this mode of measurement is with the Stop command (**STP**;), if you want, for example, to change the parameters.

1. Master initiates bus mode: **S**98;**COF**24;**ICR**0;**MSV**?0;
2. Measured values read out in cycles

	Read out AED 1	**Read out AED 2**	**Read out AED 3**	**Read out AED 4**	**Read out AED 1**
Master	**S**01;	**S**02;	**S**03;	**S**04;	**S**01;
1. AED	xxxscrlf				xxxscrlf
2. AED		xxxscrlf			
3. AED			xxxscrlf		
4. AED				xxxscrlf	

3. Master terminates bus mode: **S**98;**STP**;

At a baud rate of 38400 bd and an output rate of **ICR**0, **FMD**0 (any **ASF**), with 4 bus nodes, the cycle time is:

Cycle time 4 > = 4 ∗ 10 chars ∗ 0.3 ms

read out AED 1...4

cycle time4 > = 12 ms

At a baud rate of 38400 bd and an output rate of **ICR**0, **FMD**0 (any **ASF**), with 8 bus nodes, the cycle time is:

Cycle time 8 > = 8 ∗ 10 chars ∗ 0.3 ms

read out AED 1...8

cycle time8 > = 24 ms

5 Automatic data output

This output method is not suitable for bus mode!

If you add the decimal number 128 to the **COF**0...**COF**12 output formats, this switches the AED to continuous output mode. After a power-up or **RES** command, the AED sends out the measured values without an **MSV**? prompt. Continuous output can be deactivated with the **STP** command:

1. All the requisite settings,
2. **ICR**i; Setting the AED sampling rate
3. **COF**136; Initializing continuous output

   The AED sends binary measurement data continuously, the time interval corresponds to the **ICR**, **FMD** and **ASF** setting (see data formats and output speed)
4. **STP**; Stopping continuous transmission, first
5. **TDD**1; Saving the settings safe from power failure
6. **COF**136; The AED sends binary measurement data continuously, the time interval corresponds to the **ICR**, **FMD** and **ASF** setting (see data formats and output speed)

The next time the voltage is cut in, the AED also starts data output without waiting for a specific prompt, because this setting was stored with **TDD**1

These output formats have another special feature (depending on the trigger setting, **TRC** command):

trigger deactivated: continuous, automatic data output (**MSV**? value)

trigger activated: automatic data output only when a new measured value has been formed after triggering (**MAV**? value).

6 Notes on implementing data query in a software driver

There are certainly many software implementation options for measurement query.

One option is to use a timer to call the driver cyclically. The driver is presented using the example of a bus query (see the measurement query in bus mode section). The language is Pascal.

All the AEDs should have the same settings (**COF**, **HSM**, **FMD**, **ASF**, **ICR**).

Example:

{ **Timer interrupt e.g. every 2 ms**:

```
global variables :
Input :       boMSV          =    True, if query permitted
              Aed_adr []     =    Field index is the address,  1=AED available,
                                                               0=AED not available
              i_aed          =    Adress index for AED_ADR[]
              zusta_msv      =    Driver status
              timeout_z      =    Timeout counter, counts the timer calls

Output:       AED_MW[]       =    Index is AED address, binary data memory
              boMwZ          =    boMwZ
                                  primary processing of measured values
              boMSVerror     =    True, if timeout or query not permitted
              i_aed          =    Address index for AED_ADR[]  and AED_MW[]
              zusta_msv      =    Driver status, next status

Function:     dependent on driver status zusta_msv :
                    0   =   Send string  ‚S98;MSV?;
                    1   =   Send Sxx;
                    2   =   Wait until reception of measured value complete (4 bytes + crlf = 6 characters)
                            Store measured value, time-out monitoring,
                            presupposes 4-byte binary output (e.g. COF8)
                            Time-out time:   timeout_z  x  2 ms
}
Timer();
Var
    i1: Integer
   S1,value: String;
Begin
If  boMSV=True  then begin                         {query permitted ??}
  case  zusta_msv of
```

```
{ Start of cycle, send string  ,S98;MSV?;}
0: begin
  boMwZ:=False;                          {no valid values initially}
  ClearBuffer;                           {Clear input buffer}
  VACOMM1.WriteText(';S98;MSV?;');       {output string, the semicolon before S98, clears the AED input buffer}
  i_aed:=0;                              reset address index}
  zusta_msv:=1;                          {next status → Sxx}
  end;{case1}

{ send Sxx;, query end of cycle }
1: begin
  timeout_z:= 50;                        {applies to querying a channel, dependent on ICR, FMD, ASF}
  boMSVerror:=False;                     {no error initially}
  {find next valid AED address}
  i1:=1;
  while i1=1 do begin
     if (Aed_adr[i_aed]=1) then begin
         i1:=0;                          {valid address found, continue}
     end else begin
         i_aed:=i_aed+1;                 {next index}
         if  i_aed>31 then begin
             i1:=0;                      {continue}
             i_aed:=32;                  {invalid address}
         end;
     end; {next index}
  end; {while}
  {i_aed  indicates whether channel 0..31 active, when i_aed=32 --> inactive}
   if  i_aed>32 then begin
      {Sxx; output}
       if (i_aed<10) then begin
          Str( i_aed:1,value);
          S1:='S0'+value+';';            {'S0x;', always output address in S command as 2 digits}
       end else begin
          Str(aed_adr32:2,value);
          S1:='S'+value+';';             {'Sxx;'}
       end;
       ClearBuffer;                      {Clear input buffer}
       VACOMM1.WriteText(S1);            {Clear input buffer}
       zusta_msv:=2;                     {next status → wait for measured value}
   end else begin
      {next round S98;MSV?; as end of cycle}
       zusta_msv:=0;                     {next status → new cycle}
   end;{ i_aed
end;{case 1}
```

```
{ receiving measured values and storing in AED_MW[]}
2: begin
   i1:=VACOMM1.ReadBufUsed;                          {i1=number of bytes received}
   if i1>=6 then begin
      {characters received}
      {test for logicality: Byte4=0dh and Byte5=0ah for the end label}
       ….
       {byte0,1,2 = measured value, byte3=status}
       {storing in data memory}
        …
        AED_MW[i_aed]:= …..
        {next address index}
        i_aed:=i_aed+1;
        if  i_aed>31 then begin                      {0…31 permitted}
           zusta_msv:=0;                             {next status → new cycle}
           boMwZ:=True;                              {valid values, so end of cycle}
        end else
            zusta_msv:=1;                            {next status → Sxx; with next, possible address}
   end else begin
        {time_out check}
         timeout_z:= timeout_z-1;
         if timeout_z=0 then begin
            {timeout}
            boMSVerror:=True;                        {set error identification}
            zusta_msv:=0;                            {next status → new cycle}
         end;
   end;{timeout check}
 end;{case2}

 end; {case}
end else begin  {if  boMSV}
     {measurement query not permitted}
     boMwZ:=False;
     BoMSVerror:=True;
     Zusta_msv:=0;
end;
end; {Timer}
```

The driver has the advantage that the AED response time does not appear as a waiting time for the Master. Thanks to time_out monitoring and the test for logicality, communication that is possibly incorrect is quickly detected, without blocking the entire system. Furthermore, the bus node address configuration is of no significance (query of 1…32 AEDs possible).

Of course, this driver can also be used for 2-byte binary output or ASCII output (slight modification only, number of bytes to be received).

Index

A

D

I

M

Application document overview

Application document	**Content**
APPN001en	Digital load cells FIT® in Checkweigher applications
APPN003en	FIT®/=… - FIT®/5… - Construction and Application Conditions
APPN004en	Notes on the static adjustment of a scale with FIT® and AED
APPN005en	Measurement query via the serial link (RS232/RS485)
APPN006en	Dosing and filling with AD103 / FIT®
APPN007en	Using *AED_Panel32* program for time and frequency analysis
APPN010en	Legal for trade applications and parameter checking
APPN011en	Trigger results query
APPN012en	Commissioning CANOpen
APPN013en	Commissioning DeviceNET

I1699-1.1 en

Hottinger Baldwin Messtechnik GmbH

Postfach 100151 D-64201 Darmstadt

Im Tiefen See 45 D-64293 Darmstadt

Tel.: +49/6151/803-0 Fax: +49/6151/8039100

E-mail: support@hbm.com · www.hbm.com

measurement with confidence