

# MCA-527

Multi-Channel Analyzers

## MCA Binary Data Format

### **Exclusion of liability**

The information in this document has been carefully reviewed and is believed to be accurate and reliable. However, the GBS Elektronik GmbH assumes no liabilities for inaccuracies in this manual. This manual is subject to change without notice.

Last update: 2020-10-07

#### Address:

GBS-Elektronik GmbH  
Bautzner Landstraße 22  
01454 Großserkmannsdorf  
Tel.: (0351) 217007-0  
Fax: (0351) 217007-21

Internet: <http://www.gbs-elektronik.de>

Email to: [kontakt@gbs-elektronik.de](mailto:kontakt@gbs-elektronik.de)

## Table of Contents

1	Introduction.....	4
1.1	Basic rules.....	4
1.2	Header of the basis file block.....	4
2	General mode = 'MCA' (= 0).....	5
2.1	Basis file block.....	5
2.2	File blocks.....	9
3	General mode = 'List modes 1, 2 and 3' (= 3, 4 and 5).....	10
3.1	Basis file block.....	10
3.2	File blocks.....	12
4	General mode = 'List mode 4' (= 6).....	13
4.1	Basis file block.....	13
4.2	File blocks.....	15

# 1 Introduction

The MCA binary data format has been developed by us for the present applications of the MCA527, however, it is also designed to be extended for future applications or MCA types. The MCA527 prefers the MCA binary data format since it is more compact and easier to write than the MCA spectral data format<sup>1</sup> (\*.spe). All MCA527's with microSD memory card socket and firmware version 12.02 or higher are able to write measurement data to MCA binary data files (\*.mca).

The MCA binary data format is not exclusively used by the MCA527 but also by computer programs because the compactness of the file format is ideally suited for large amounts of data.

## 1.1 Basic rules

- MCA binary data files consist of one or several file blocks.
- For reasons of simplification, the MCA527 only writes file blocks that size is always a multiple of 512. The rest of each file block is filled with dummy bytes. This rule does not refer to files that are written by computer programs.
- The first file block is named basis file block. It starts with a 28 bytes long header. The meaning of the further bytes of the basis file block is depending on the general mode.
- Computer programs can add freely defined file blocks behind the regular file blocks. Each of these file blocks must start with an unsigned long value that defines the file block size including this 4 bytes. The computer programs are free to define the file block format according to its needs.

## 1.2 Header of the basis file block

Position	Name	Data type	Remarks
Byte offset 0	File identification	14 characters	The string 'MCA527BINARY ' identifies the file as a binary data file that has been directly written by the MCA527.  The string 'MCA527BIN_APP ' identifies the file as binary data file that has been written by any application. The data origin is the MCA527.
Byte offset 14	Used bytes of the basis file block	unsigned short	This value defines the number of valid bytes <sup>2</sup> .  If the file has been directly written by the MCA527, the rest of the basis file block is filled with dummy bytes to get a multiple of 512.
Byte offset 16	Firmware version	unsigned short	
Byte offset 18	Hardware version	unsigned short	
Byte offset 20	Firmware modification	unsigned short	
Byte offset 22	Hardware modification	unsigned short	
Byte offset 24	Serial number	unsigned short	
Byte offset 26	General mode	unsigned short	

<sup>1</sup> See document "MCA Data Spectral Format" (MCA\_Spectral\_Data\_Format\_XXXX\_XX\_XX.pdf).

<sup>2</sup> Applications which read files from this type should take into consideration that the number of valid bytes are potentially increased by higher firmware or software versions.

## 2 General mode = 'MCA' (= 0)

In this general mode the MCA527 runs as an ordinary multi-channel analyzer.

### 2.1 Basis file block

The following table shows the basis file block of a MCA527 with general mode equal 'MCA' (=0).

Position	Name	Data type	References <sup>3</sup>
Byte offset 0	Header	28 bytes	
Byte offset 28	MCA acquire mode	unsigned short	CMD_QUERY_STATE, byte offset 0
Byte offset 30	MCA channels	unsigned short	CMD_QUERY_STATE, byte offset 36
Byte offset 32	LLD	unsigned short	CMD_QUERY_STATE, byte offset 40
Byte offset 34	ULD	unsigned short	CMD_QUERY_STATE, byte offset 42
Byte offset 36	Threshold [ * 0,1 %]	unsigned short	CMD_QUERY_STATE527, byte offset 66
Byte offset 38	Preset	unsigned short	CMD_QUERY_STATE, byte offset 2
Byte offset 40	Preset value	unsigned long	CMD_QUERY_STATE, byte offset 4
Byte offset 44	Preset ROI begin	unsigned short	CMD_QUERY_STATE, byte offset 44
Byte offset 46	Preset ROI end	unsigned short	CMD_QUERY_STATE, byte offset 46
Byte offset 48	MCS channels	unsigned short	CMD_QUERY_STATE, byte offset 92
Byte offset 50	MCS input	unsigned short	CMD_QUERY_STATE, byte offset 84
Byte offset 52	MCS time per channel [ * 0.1 msec]	unsigned long	CMD_QUERY_STATE527, byte offset 92
Byte offset 56	Stabilisation state	unsigned short	CMD_QUERY_STATE, byte offset 68
Byte offset 58	Stabilisation result	unsigned short	CMD_QUERY_STATE, byte offset 70
Byte offset 60	Stabilisation ROI begin	unsigned short	CMD_QUERY_STATE, byte offset 72
Byte offset 62	Stabilisation ROI end	unsigned short	CMD_QUERY_STATE, byte offset 74
Byte offset 64	Stabilisation counter	unsigned long	CMD_QUERY_SYSTEM_DATA, byte offset 80
Byte offset 68	Stabilisation offset	long	CMD_QUERY_SYSTEM_DATA, byte offset 84
Byte offset 72	Stabilisation offset minimum	long	CMD_QUERY_SYSTEM_DATA, byte offset 88
Byte offset 76	Stabilisation offset maximum	long	CMD_QUERY_SYSTEM_DATA, byte offset 92
Byte offset 80	Stabilisation area preset	unsigned long	CMD_QUERY_SYSTEM_DATA, byte offset 116
Byte offset 84	Stabilisation time preset [sec]	unsigned short	CMD_QUERY_SYSTEM_DATA, byte offset 120
Byte offset 86	Repeat value	unsigned short	CMD_QUERY_STATE, byte offset 12
Byte offset 88	Amplifier coarse gain	unsigned short	CMD_QUERY_STATE, byte offset 48
Byte offset 90	Amplifier fine gain	unsigned short	CMD_QUERY_STATE, byte offset 50
Byte offset 92	ADC input	unsigned short	CMD_QUERY_STATE, byte offset 76
Byte offset 94	ADC input polarity	unsigned short	CMD_QUERY_STATE, byte offset 78
Byte offset 96	High voltage [V]	unsigned short	CMD_QUERY_STATE, byte offset 56
Byte offset 98	High voltage polarity	unsigned short	CMD_QUERY_STATE, byte offset 58
Byte offset 100	HV inhibit mode	short	CMD_QUERY_STATE, byte offset 122

<sup>3</sup> The parameters starting from byte offset 16 refer to parameters returned by MCA query commands. The table cells contain the name of the command and the byte offset within the result data array. See the document "Description of the MCA527 Firmware Commands" (MCA527\_Firmware\_Commands\_XXXX\_XX\_XX.pdf).

Position	Name	Data type	References
Byte offset 102	Preamplifier power switches	unsigned short	CMD_QUERY_STATE, byte offset 60
Byte offset 104	PZC value	unsigned short	CMD_QUERY_STATE, byte offset 62
Byte offset 106	Low shaping time [ * 0.1 $\mu$ sec]	unsigned char	CMD_QUERY_SYSTEM_DATA, byte offset 122
Byte offset 107	High shaping time [ * 0.1 $\mu$ sec]	unsigned char	CMD_QUERY_SYSTEM_DATA, byte offset 123
Byte offset 108	Shaping time choice	unsigned short	CMD_QUERY_STATE, byte offset 80
Byte offset 110	Pile up rejection (PUR) state	unsigned short	CMD_QUERY_STATE, byte offset 82
Byte offset 112	Trigger filter for low shaping time	unsigned char	CMD_QUERY_STATE527, byte offset 34
Byte offset 113	Trigger filter for high shaping time	unsigned char	CMD_QUERY_STATE527, byte offset 35
Byte offset 114	Offset DAC	unsigned short	CMD_QUERY_STATE527, byte offset 38
Byte offset 116	Flattop time [ * 0.1 $\mu$ sec]	unsigned short	CMD_QUERY_STATE527, byte offset 74
Byte offset 118	Trigger level for automatic threshold calculation [ * 0.0625]	unsigned short	CMD_QUERY_STATE527, byte offset 78
Byte offset 120	Evaluation filter type	unsigned short	CMD_QUERY_STATE527, byte offset 72
Byte offset 122	Jitter correction	unsigned char	CMD_QUERY_STATE527, byte offset 114
Byte offset 123	Baseline restoring	unsigned char	CMD_QUERY_STATE527, byte offset 115
Byte offset 124	Gating mode	unsigned char	CMD_QUERY_STATE527, byte offset 122
Byte offset 125	Gating polarity	unsigned char	CMD_QUERY_STATE527, byte offset 123
Byte offset 126	Gating shift	unsigned char	CMD_QUERY_STATE527, byte offset 124
Byte offset 127	unused	1 byte	
Byte offset 128	TTL low level [ * 0.1 V]	unsigned char	CMD_QUERY_STATE527_EX, byte offset 96
Byte offset 129	TTL high level [ * 0.1 V]	unsigned char	CMD_QUERY_STATE527_EX, byte offset 97
Byte offset 130	Trigger level for automatic threshold calculation for direct input [ * 0.0625]	unsigned short	CMD_QUERY_STATE527_EX, byte offset 98
Byte offset 132	Extension port part A configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 24
Byte offset 133	Extension port part B configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 25
Byte offset 134	Extension port part C configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 26
Byte offset 135	Extension port part D configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 27
Byte offset 136	Extension port part E configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 28
Byte offset 137	Extension port part F configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 29
Byte offset 138	Extension port parts availability	unsigned char	CMD_QUERY_STATE527_EX, byte offset 30
Byte offset 139	Extension port polarity flags	unsigned char	CMD_QUERY_STATE527_EX, byte offset 32
Byte offset 140	Extension port pulser 1 period	unsigned long	CMD_QUERY_STATE527_EX, byte offset 36
Byte offset 144	Extension port pulser 2 period	unsigned long	CMD_QUERY_STATE527_EX, byte offset 40
Byte offset 148	Extension port pulser 1 width	unsigned long	CMD_QUERY_STATE527_EX, byte offset 44
Byte offset 152	Extension port pulser 2 width	unsigned long	CMD_QUERY_STATE527_EX, byte offset 48
Byte offset 156	Extension port RS232 baud rate	unsigned short	CMD_QUERY_STATE527_EX, byte offset 52
Byte offset 158	Extension port RS232 flags	unsigned short	CMD_QUERY_STATE527_EX, byte offset 54
Byte offset 160	Extension port counter 1	unsigned long	CMD_QUERY_STATE527_EX, byte offset 56
Byte offset 164	Extension port counter 2	unsigned long	CMD_QUERY_STATE527_EX, byte offset 68
Byte offset 168	User data size [ * 512 Bytes]	unsigned short	

Position	Name	Data type	References
Byte offset 170	Start flag	unsigned short	CMD_QUERY_STATE, byte offset 130
Byte offset 172	Start time	unsigned long	CMD_QUERY_STATE, byte offset 100
Byte offset 176	Real time [sec]	unsigned long	CMD_QUERY_STATE, byte offset 20
Byte offset 180	Dead time [msec]	unsigned long	CMD_QUERY_STATE, byte offset 28
Byte offset 184	Fast dead time [msec]	unsigned long	CMD_QUERY_STATES527, byte offset 68
Byte offset 188	Detected counts	64 bit integer	CMD_QUERY_SYSTEM_DATA, byte offset 10
Byte offset 196	PUR counter	unsigned long	CMD_QUERY_STATES527_EX, byte offset 20
Byte offset 200	Battery current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 64
Byte offset 204	Charger current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 116
Byte offset 208	HV primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 68
Byte offset 212	+12V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 72
Byte offset 216	-12V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 76
Byte offset 220	+24V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 80
Byte offset 224	-24V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 84
Byte offset 228	Battery voltage [mV] at stop	unsigned long	CMD_QUERY_POWER, byte offset 88
Byte offset 232	High voltage [ * 1.2 V] at stop	unsigned long	CMD_QUERY_POWER, byte offset 92
Byte offset 236	+12V actual value [ * 0.0625 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 100
Byte offset 237	-12V actual value [ * 0.0625 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 101
Byte offset 238	+24V actual value [ * 0.125 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 102
Byte offset 239	-24V actual value [ * 0.125 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 103
Byte offset 240	Voltage on SUB-D9 pin3 [ * 0.3125 mV] at stop	unsigned short	CMD_QUERY_POWER, byte offset 104
Byte offset 242	Voltage on SUB-D9 pin5 [ * 0.3125 mV] at stop	unsigned short	CMD_QUERY_POWER, byte offset 114
Byte offset 244	Current source state on SUB-D9 pin5	unsigned short	CMD_QUERY_POWER, byte offset 58
Byte offset 246	Current source value on SUB-D9 pin5 [ * 0.1 $\mu$ A]	unsigned short	CMD_QUERY_POWER, byte offset 56
Byte offset 248	Input resistance on SUB-D9 pin5 [k $\Omega$ ]	unsigned short	CMD_QUERY_POWER, byte offset 60
Byte offset 250	ADC correction offset on SUB-D9 pin5 [LSB]	char	CMD_QUERY_POWER, byte offset 62
Byte offset 251	Gain correction factor on SUB-D9 pin5	char	CMD_QUERY_POWER, byte offset 63
Byte offset 252	ADC correction offset on SUB-D9 pin3 [LSB]	char	CMD_QUERY_POWER, byte offset 96
Byte offset 253	Gain correction factor on SUB-D9 pin3	char	CMD_QUERY_POWER, byte offset 97
Byte offset 254	MCA temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATES527, byte offset 80
Byte offset 256	Detector temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATES527, byte offset 82
Byte offset 258	Power module temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATES527, byte offset 104
<i>The following parameters are written since firmware version 14.02.</i>			
Byte offset 260	Time window 0 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATES527_EX2, byte offset 44
Byte offset 264	Time window 1 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATES527_EX2, byte offset 48

Position	Name	Data type	References
Byte offset 268	Time window 2 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 52
Byte offset 272	Time window 3 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 56
Byte offset 276	Time window 4 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 60
Byte offset 280	Time window 5 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 64
Byte offset 284	Time window 6 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 68
Byte offset 288	Time window 7 width for gating mode 'sort by time'	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 72
Byte offset 292	Core clock [ * 100 MHz]	unsigned short	CMD_QUERY_STATE527, byte offset 32
<i>The following parameters are written since firmware version 14.03.</i>			
Byte offset 294	Fractional digits of the real time [msec]	unsigned short	CMD_QUERY_STATE527_EX, byte offset 82
<i>The following parameters are written since firmware version 16.00.</i>			
Byte offset 296	Counts outside the spectrum	64 bit integer	CDM_QUERY_SYSTEM_DATA, byte offset 18
Byte offset 304	ADC sample rate [ * 1 kHz]	unsigned short	CDM_QUERY_SYSTEM_DATA, byte offset 130
Byte offset 306	MCS time per channel for gating mode 'sort by time'	unsigned short	CMD_QUERY_STATE527_EX2, byte offset 42
Byte offset 308	Dummy bytes to get a multiple of 512.	204 bytes	



## 2.2 File blocks

<b>Basis file block</b>	This file block always exists. It is 512 bytes long. See above.
<b>User data</b>	This file block always exists. It contains the user data of the MCA527. The block size corresponds to the 'User data size' (basis file block, byte offset 168).
<b>MCS spectrum or</b>  <b>MCS spectrum</b> <b>(Gating mode = 'Sort by time')</b>	<p>This file block exists if the 'MCA acquire mode' (basis file block, byte offset 28) is 'MCS' (=1). The block size is 4 * 'MCS channels' (basis file block, byte offset 48). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.</p> <p>This file block exists since firmware version 16.00 if the 'Gating mode' (basis file block, byte offset 124) is 'Sort by time' (=3). The block size is 4 * 'MCS channels' (basis file block, byte offset 48). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.</p>
<b>MCS spectrum (gated)</b>	This file block exists if the 'MCA acquire mode' (basis file block, byte offset 28) is 'MCS' (=1) and the 'Gating mode' (basis file block, byte offset 124) is 'Sort' (=2). The block size is 4 * MCS channels (basis file block, byte offset 48). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.
<b>MCS spectrum (counter 1)</b>	This file block exists if the 'MCA acquire mode' (basis file block, byte offset 28) is 'MCS' (=1) and the 'Extension port part E configuration' (basis file block, byte offset 136) is 'Counter' (=1). The block size is 4 * MCS channels (basis file block, byte offset 48). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.
<b>MCS spectrum (counter 2)</b>	This file block exists if the 'MCA acquire mode' (basis file block, byte offset 28) is 'MCS' (=1) and the 'Extension port part C configuration' (basis file block, byte offset 134) is 'Counter' (=1). The block size is 4 * MCS channels (basis file block, byte offset 48). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.
<b>MCA spectrum or</b>  <b>MCA spectrum (time window 0)</b>	<p>This file block exists if either the 'MCA acquire mode' (basis file block, byte offset 28) is 'MCA' (=0), or it is 'MCS' (=1) and 'MCS input' (basis file block, byte offset 50) is 'Input Rate' (=1) or 'LLD/ULD' (=2).</p> <p>If the 'Gating mode' (basis file block, byte offset 124) is 'Sort by time' (=3), this MCA spectrum refers to time window 0.</p> <p>The block size is 4 * 'MCA channels' (basis file block, byte offset 30). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.</p>
<b>MCA spectrum (rejected) or</b>  <b>MCA spectrum (time window 1)</b>	<p>This file block exists if the conditions for the existence of the MCA spectrum are met and the 'Gating mode' (basis file block, byte offset 124) is 'Sort by state' (=2).</p> <p>If the 'Gating mode' (basis file block, byte offset 124) is 'Sort by time' (=3), this MCA spectrum refers to time window 1. It exists only if the time window 0 (basis file block, byte offset 260) is not set to infinity.</p> <p>The block size is 4 * 'MCA channels' (basis file block, byte offset 30). If the calculated size is not a multiple of 512, the file block is filled with dummy bytes.</p>
<b>MCA spectra</b> <b>(time windows 2 ... 7)</b>	<p>If the 'Gating mode' (basis file block, byte offset 124) is 'Sort by time' (=3), these MCA spectra refer to time windows 2 ... 7. They exist only if no previous time window is set to infinity.</p> <p>The block size is always 4 * 'MCA channels' (basis file block, byte offset 30). If the calculated size is not a multiple of 512, the file blocks are filled with dummy bytes.</p>
<b>RS232 received data</b>	This file block exists if 'Extension port part A configuration' (basis file block, byte offset 132) is 'RS232 (TX/RX, with data buffering)' (=5) or 'Extension port part C configuration' (basis file block, byte offset 134) is 'RS232 (RX, with data buffering)' (=5). The file block is always 1024 bytes long.

### 3 General mode = 'List modes 1, 2 and 3' (= 3, 4 and 5)

This file format is used by the 'WinTimestamps' computer program that operates the list mode 1 (level triggered, general mode = 3), the list mode 2 (edge triggered), general mode = 4) and the list mode 3 (analog high rate counting, general mode = 5). Since the parameters of these general modes differ only slightly, an identical file format is used for these general modes. The basis file block contains however some parameters that are only relevant for one of these general modes.

#### 3.1 Basis file block

Position	Name	Data type	References
Byte offset 0	Header	28 bytes	
Byte offset 28	Application identification	32 characters	'WinTimestamps Version 01.00.0000'
Byte offset 60	Time unit length [nanoseconds]	unsigned short	
Byte offset 62	Preset	unsigned short	CMD_QUERY_STATE, byte offset 2
Byte offset 64	Preset value	unsigned long	CMD_QUERY_STATE, byte offset 4
Byte offset 68	Preset memory size	unsigned long	CMD_QUERY_STATE527_EX, byte offset 4
Byte offset 72	Used memory size	unsigned long	CMD_QUERY_STATE527_EX, byte offset 8
Byte offset 76	High voltage [V]	unsigned short	CMD_QUERY_STATE, byte offset 56
Byte offset 78	High voltage polarity	unsigned short	CMD_QUERY_STATE, byte offset 58
Byte offset 80	HV inhibit mode	short	CMD_QUERY_STATE, byte offset 122
Byte offset 82	Preamplifier power switches	unsigned short	CMD_QUERY_STATE, byte offset 60
Byte offset 84	TTL low level [ * 0.1 V] <sup>4</sup>	unsigned char	CMD_QUERY_STATE527_EX, byte offset 96
Byte offset 85	TTL high level [ * 0.1 V] <sup>4</sup>	unsigned char	CMD_QUERY_STATE527_EX, byte offset 97
Byte offset 86	Amplifier coarse gain <sup>5</sup>	unsigned short	CMD_QUERY_STATE, byte offset 48
Byte offset 88	ADC input polarity <sup>5</sup>	unsigned short	CMD_QUERY_STATE, byte offset 78
Byte offset 90	Shaping time choice <sup>5</sup>	unsigned short	CMD_QUERY_STATE, byte offset 80
Byte offset 92	Trigger filter for low shaping time <sup>5</sup>	unsigned char	CMD_QUERY_STATE527, byte offset 34
Byte offset 93	Trigger filter for high shaping time <sup>5</sup>	unsigned char	CMD_QUERY_STATE527, byte offset 35
Byte offset 94	Offset DAC <sup>5</sup>	unsigned short	CMD_QUERY_STATE527, byte offset 38
Byte offset 96	Trigger level for automatic threshold calculation [ * 0.0625] <sup>5</sup>	unsigned short	CMD_QUERY_STATE527, byte offset 78
Byte offset 98	Set trigger threshold [ * 0.00006103515625] <sup>5</sup>	long	CMD_QUERY_STATE527, byte offset 116
Byte offset 102	Extension port part A configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 24
Byte offset 103	Extension port part B configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 25
Byte offset 104	Extension port part C configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 26
Byte offset 105	Extension port part F configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 29
Byte offset 106	Extension port RS232 baud rate	unsigned short	CMD_QUERY_STATE527_EX, byte offset 52
Byte offset 108	Extension port RS232 flags	unsigned short	CMD_QUERY_STATE527_EX, byte offset 54
Byte offset 110	Start flag	unsigned short	CMD_QUERY_STATE, byte offset 130

4 relevant only for general mode 'Timestamps recorder (level triggered)' (= 3)

5 relevant only for general mode 'Timestamps recorder (edge triggered)' (= 4)

Position	Name	Data type	References
Byte offset 112	Start time	unsigned long	CMD_QUERY_STATE, byte offset 100
Byte offset 116	Real time [sec]	unsigned long	CMD_QUERY_STATE, byte offset 20
Byte offset 120	Battery current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 64
Byte offset 124	Charger current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 116
Byte offset 128	HV primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 68
Byte offset 132	+12V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 72
Byte offset 136	-12V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 76
Byte offset 140	+24V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 80
Byte offset 144	-24V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 84
Byte offset 148	Battery voltage [mV] at stop	unsigned long	CMD_QUERY_POWER, byte offset 88
Byte offset 152	High voltage [ * 1.2 V] at stop	unsigned long	CMD_QUERY_POWER, byte offset 92
Byte offset 156	+12V actual value [ * 0.0625 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 100
Byte offset 157	-12V actual value [ * 0.0625 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 101
Byte offset 158	+24V actual value [ * 0.125 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 102
Byte offset 159	-24V actual value [ * 0.125 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 103
Byte offset 160	Voltage on SUB-D9 pin3 [ * 0.3125 mV] at stop	unsigned short	CMD_QUERY_POWER, byte offset 104
Byte offset 162	Voltage on SUB-D9 pin5 [ * 0.3125 mV] at stop	unsigned short	CMD_QUERY_POWER, byte offset 114
Byte offset 164	Current source state on SUB-D9 pin5	unsigned short	CMD_QUERY_POWER, byte offset 58
Byte offset 166	Current source value on SUB-D9 pin5 [ * 0.1 $\mu$ A]	unsigned short	CMD_QUERY_POWER, byte offset 56
Byte offset 168	Input resistance on SUB-D9 pin5 [k $\Omega$ ]	unsigned short	CMD_QUERY_POWER, byte offset 60
Byte offset 170	ADC correction offset on SUB-D9 pin5 [LSB]	char	CMD_QUERY_POWER, byte offset 62
Byte offset 171	Gain correction factor on SUB-D9 pin5	char	CMD_QUERY_POWER, byte offset 63
Byte offset 172	ADC correction offset on SUB-D9 pin3 [LSB]	char	CMD_QUERY_POWER, byte offset 96
Byte offset 173	Gain correction factor on SUB-D9 pin3	char	CMD_QUERY_POWER, byte offset 97
Byte offset 174	MCA temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATES527, byte offset 80
Byte offset 176	Detector temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATES527, byte offset 82
Byte offset 178	Power module temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATES527, byte offset 104
<i>The following parameters are written by WinTimestamps since version 1.00.0007.</i>			
Byte offset 180	Repeat mode	char	
Byte offset 181	Repeat mode options	char	
Byte offset 182	Repeat value	short	CMD_QUERY_STATE, byte offset 12
<i>The following parameters are written by WinTimestamps since version 1.01.0000.</i>			
Byte offset 184	AHRC group 0 width	unsigned long	CMD_QUERY_STATES527_EX2, byte offset 0
Byte offset 188	AHRC group 1 width	unsigned long	CMD_QUERY_STATES527_EX2, byte offset 4
Byte offset 192	AHRC group 2 width	unsigned long	CMD_QUERY_STATES527_EX2, byte offset 8

Position	Name	Data type	References
Byte offset 196	AHRC group 3 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 12
Byte offset 200	AHRC group 4 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 16
Byte offset 204	AHRC group 5 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 20
Byte offset 208	AHRC group 6 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 24
Byte offset 212	AHRC group 7 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 28
Byte offset 216	AHRC group 8 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 32
Byte offset 220	AHRC group 9 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 36
Byte offset 224	AHRC trigger threshold	unsigned short	CMD_QUERY_STATE527_EX2, byte offset 40
Byte offset 226	Time coding method <sup>6</sup>	unsigned short	

## 3.2 File blocks

<b>Basis file block</b>	This file block always exists. See above.
<b>Timestamps</b>	<p>The file block size corresponds to the used memory size (basis file block, byte offset 72).</p> <p>There are three time coding methods. The program automatically uses the time coding method with the minimal storage requirements.</p> <p><b>Time coding method 0:</b> The values are stored with varying byte length. At this method the bytes are arranged in big-endian order. The first byte always contains the code for the byte length of the value.</p> <pre> 1 byte:  XXXXXXXX                               0 ... 191 2 byte:  11XXXXXX XXXXXXXX                       192 ... 12,479 3 byte:  1111XXXX XXXXXXXX XXXXXXXX             12,480 ... 798,911 4 byte:  111111XX XXXXXXXX XXXXXXXX XXXXXXXX   798,912 ... 67,907,775 </pre> <p>Each value represents the number of time units which are bygone since the previous event. The time unit length is defined in the basis file block. Values of 67,907,775 means that no event has occurred for 67,907,775 time units. Only the next value less than 67,907,775 marks the next event.</p> <p><b>Time coding method 1:</b> Each unsigned int8 value represents the number of time units which are bygone since the previous event. The time unit length is defined in the basis file block. Values of 255 means that no event has occurred for 255 time units. Only the next value less than 255 marks the next event.</p> <p><b>Time coding method 2:</b> Each unsigned int16 value represents the number of time units which are bygone since the previous event. The time unit length is defined in the basis file block. Values of 65,535 means that no event has occurred for 65,535 time units. Only the next value less than 65,535 marks the next event.</p>
<b>RS232 received data</b>	This file block exists if the used memory size (basis file block, byte offset 72) is greater than zero <sup>7</sup> , and 'Extension port part A configuration' (basis file block, byte offset 102) is 'RS232 (TX/RX, with data buffering)' (=5) or 'Extension port part C configuration' (basis file block, byte offset 104) is 'RS232 (RX, with data buffering)' (=5). The file block is always 1024 bytes long.
<b>Remarks</b>	This file block is optional. It is a freely defined file block.

<sup>6</sup> If this parameter is missing in the file, time coding method 2 is assumed.

<sup>7</sup> If the used memory size is equal to zero, the file contains no measurement data, and consequently also no RS232 received data.

## 4 General mode = 'List mode 4' (= 6)

This file format is used by the 'MCA-527 List Mode 4' software library that operates this general mode.

### 4.1 Basis file block

Position	Name	Data type	References
Byte offset 0	Header	28 bytes	
Byte offset 28	Application identification	32 characters	'Mca527Im4.dll Version 01.00.0000'
Byte offset 60	Time unit length [nanoseconds]	unsigned short	
Byte offset 62	Preset	unsigned short	CMD_QUERY_STATE, byte offset 2
Byte offset 64	Preset value	unsigned long	CMD_QUERY_STATE, byte offset 4
Byte offset 68	Preset memory size	unsigned long	CMD_QUERY_STATE527_EX, byte offset 4
Byte offset 72	Used memory size	unsigned long	CMD_QUERY_STATE527_EX, byte offset 8
Byte offset 76	High voltage [V]	unsigned short	CMD_QUERY_STATE, byte offset 56
Byte offset 78	High voltage polarity	unsigned short	CMD_QUERY_STATE, byte offset 58
Byte offset 80	HV inhibit mode	short	CMD_QUERY_STATE, byte offset 122
Byte offset 82	Preamplifier power switches	unsigned short	CMD_QUERY_STATE, byte offset 60
Byte offset 84	Amplifier coarse gain	unsigned short	CMD_QUERY_STATE, byte offset 48
Byte offset 86	ADC input polarity	unsigned short	CMD_QUERY_STATE, byte offset 78
Byte offset 88	Shaping time choice	unsigned short	CMD_QUERY_STATE, byte offset 80
Byte offset 90	Trigger filter for low shaping time	unsigned char	CMD_QUERY_STATE527, byte offset 34
Byte offset 91	Trigger filter for high shaping time	unsigned char	CMD_QUERY_STATE527, byte offset 35
Byte offset 92	Offset DAC	unsigned short	CMD_QUERY_STATE527, byte offset 38
Byte offset 94	Trigger level for automatic threshold calculation [ * 0.0625]	unsigned short	CMD_QUERY_STATE527, byte offset 78
Byte offset 98	Set trigger threshold [ * 0.00006103515625]	long	CMD_QUERY_STATE527, byte offset 116
Byte offset 100	Extension port part A configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 24
Byte offset 101	Extension port part B configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 25
Byte offset 102	Extension port part C configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 26
Byte offset 103	Extension port part D configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 27
Byte offset 104	Extension port part E configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 28
Byte offset 105	Extension port part F configuration	unsigned char	CMD_QUERY_STATE527_EX, byte offset 29
Byte offset 106	Extension port parts availability	unsigned char	CMD_QUERY_STATE527_EX, byte offset 30
Byte offset 107	Extension port polarity flags	unsigned char	CMD_QUERY_STATE527_EX, byte offset 32
Byte offset 108	Extension port pulser 1 period	unsigned long	CMD_QUERY_STATE527_EX, byte offset 36
Byte offset 112	Extension port pulser 2 period	unsigned long	CMD_QUERY_STATE527_EX, byte offset 40
Byte offset 116	Extension port pulser 3 period	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 76
Byte offset 120	Extension port pulser 1 width	unsigned long	CMD_QUERY_STATE527_EX, byte offset 44
Byte offset 124	Extension port pulser 2 width	unsigned long	CMD_QUERY_STATE527_EX, byte offset 48
Byte offset 128	Extension port pulser 3 width	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 80
Byte offset 132	Extension port RS232 baud rate	unsigned short	CMD_QUERY_STATE527_EX, byte offset 52

Position	Name	Data type	References
Byte offset 134	Extension port RS232 flags	unsigned short	CMD_QUERY_STATE527_EX, byte offset 54
Byte offset 136	Extension port counter 1 at stop	unsigned long	CMD_QUERY_STATE527_EX, byte offset 64
Byte offset 140	Extension port counter 2 at stop	unsigned long	CMD_QUERY_STATE527_EX, byte offset 76
Byte offset 144	Extension port counter 3 at stop	unsigned long	CMD_QUERY_STATE527_EX2, byte offset 92
Byte offset 148	Start flag	unsigned short	CMD_QUERY_STATE, byte offset 130
Byte offset 150	Fast trigger input	unsigned short	CMD_QUERY_STATE527_EX2, byte offset 130
Byte offset 152	Start time	unsigned long	CMD_QUERY_STATE, byte offset 100
Byte offset 156	Real time [sec]	unsigned long	CMD_QUERY_STATE, byte offset 20
Byte offset 160	Battery current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 64
Byte offset 164	Charger current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 116
Byte offset 168	HV primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 68
Byte offset 172	+12V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 72
Byte offset 176	-12V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 76
Byte offset 180	+24V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 80
Byte offset 184	-24V primary current [mA] at stop	unsigned long	CMD_QUERY_POWER, byte offset 84
Byte offset 188	Battery voltage [mV] at stop	unsigned long	CMD_QUERY_POWER, byte offset 88
Byte offset 192	High voltage [ * 1.2 V] at stop	unsigned long	CMD_QUERY_POWER, byte offset 92
Byte offset 196	+12V actual value [ * 0.0625 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 100
Byte offset 197	-12V actual value [ * 0.0625 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 101
Byte offset 198	+24V actual value [ * 0.125 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 102
Byte offset 199	-24V actual value [ * 0.125 V] at stop	unsigned char	CMD_QUERY_POWER, byte offset 103
Byte offset 200	Voltage on SUB-D9 pin3 [ * 0.3125 mV] at stop	unsigned short	CMD_QUERY_POWER, byte offset 104
Byte offset 202	Voltage on SUB-D9 pin5 [ * 0.3125 mV] at stop	unsigned short	CMD_QUERY_POWER, byte offset 114
Byte offset 204	Current source state on SUB-D9 pin5	unsigned short	CMD_QUERY_POWER, byte offset 58
Byte offset 206	Current source value on SUB-D9 pin5 [ * 0.1 $\mu$ A]	unsigned short	CMD_QUERY_POWER, byte offset 56
Byte offset 208	Input resistance on SUB-D9 pin5 [k $\Omega$ ]	unsigned short	CMD_QUERY_POWER, byte offset 60
Byte offset 210	ADC correction offset on SUB-D9 pin5 [LSB]	char	CMD_QUERY_POWER, byte offset 62
Byte offset 211	Gain correction factor on SUB-D9 pin5	char	CMD_QUERY_POWER, byte offset 63
Byte offset 212	ADC correction offset on SUB-D9 pin3 [LSB]	char	CMD_QUERY_POWER, byte offset 96
Byte offset 213	Gain correction factor on SUB-D9 pin3	char	CMD_QUERY_POWER, byte offset 97
Byte offset 214	MCA temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATE527, byte offset 80
Byte offset 216	Detector temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATE527, byte offset 82
Byte offset 218	Power module temperature [ * 0.0078125 $^{\circ}$ C] at stop	short	CMD_QUERY_STATE527, byte offset 104
Byte offset 220	ADC pipeline latency	unsigned char	CMD_QUERY_SYSTEM_DATA, byte offset 104
Byte offset 221	Time coding method	unsigned short	

## 4.2 File blocks

<b>Basis file block</b>	This file block always exists. See above.																														
<b>List entries</b>	<p>The file block size corresponds to the used memory size (basis file block, byte offset 72).</p> <p>In principle, each list entry is an event information followed by a time information. The event information is either the calculated channel number within a 16k<sup>8</sup> spectrum or the information about events that do not result in a count within the spectrum.</p> <p>If the event information is a channel number, it consists of two bytes: 0XXXXXXXXXXXXXXXXX. The bytes are arranged in big-endian order. Bit 15 is always 0. Bit 14 has not been used so far. It should be 0. Bit 13 ... 0 stand for the calculated channel number.</p> <p>All other event information consists of one byte in the format 10XXXXXXXX.</p> <table border="0"> <tr> <td>10000000</td> <td>Event above the spectrum range.</td> </tr> <tr> <td>10000001</td> <td>Event below the spectrum range.</td> </tr> <tr> <td>10000010</td> <td>An event was triggered, but could not be evaluated because of pile up rejection.</td> </tr> <tr> <td>10000011</td> <td>An event was triggered, but could not be evaluated by the jitter correction algorithm.</td> </tr> <tr> <td>10000100</td> <td>An event was triggered, but could not be evaluated because of a subsequent event.</td> </tr> <tr> <td>10000101</td> <td>Beginning of an ADC overflow or underflow.</td> </tr> <tr> <td>10000110</td> <td>End of an ADC overflow or underflow<sup>9</sup>.</td> </tr> <tr> <td>10000111</td> <td>Beginning of a discarded cycle<sup>10</sup>.</td> </tr> <tr> <td>10001000</td> <td>Stop of the measurement because of reaching the preset real time.</td> </tr> </table> <p>Besides, there is a single byte long non-event information in the format 11XXXXXXXX. A non-event information stands for a time interval without events that cannot be coded with the used time coding method. There are three time coding methods. The program automatically uses the time coding method with the minimal storage requirements.</p> <p><b>Time coding method 0:</b> The time information is stored with varying byte length. At this method the bytes are arranged in big-endian order. The first byte always contains the code for the byte length of the value.</p> <table border="0"> <tr> <td>1 byte:</td> <td>XXXXXXXX</td> <td>0 ... 191</td> </tr> <tr> <td>2 byte:</td> <td>11XXXXXXXX XXXXXXXX</td> <td>192 ... 12,479</td> </tr> <tr> <td>3 byte:</td> <td>1111XXXX XXXXXXXX XXXXXXXX</td> <td>12,480 ... 798,911</td> </tr> <tr> <td>4 byte:</td> <td>111111XX XXXXXXXX XXXXXXXX XXXXXXXX</td> <td>798,912 ... 67,907,775</td> </tr> </table> <p>Each value represents the number of time units which are bygone since the previous event. The time unit length is defined in the basis file block. Time intervals that exceeds the value of 67,907,775 are coded as non-event information. 11000000 means that no event has occurred for 67,907,776 time units.</p> <p>Time coding method 0 is the native method of the MCA527. It is used as long as the alternative coding methods do not save more than 10 percent disk space.</p>	10000000	Event above the spectrum range.	10000001	Event below the spectrum range.	10000010	An event was triggered, but could not be evaluated because of pile up rejection.	10000011	An event was triggered, but could not be evaluated by the jitter correction algorithm.	10000100	An event was triggered, but could not be evaluated because of a subsequent event.	10000101	Beginning of an ADC overflow or underflow.	10000110	End of an ADC overflow or underflow <sup>9</sup> .	10000111	Beginning of a discarded cycle <sup>10</sup> .	10001000	Stop of the measurement because of reaching the preset real time.	1 byte:	XXXXXXXX	0 ... 191	2 byte:	11XXXXXXXX XXXXXXXX	192 ... 12,479	3 byte:	1111XXXX XXXXXXXX XXXXXXXX	12,480 ... 798,911	4 byte:	111111XX XXXXXXXX XXXXXXXX XXXXXXXX	798,912 ... 67,907,775
10000000	Event above the spectrum range.																														
10000001	Event below the spectrum range.																														
10000010	An event was triggered, but could not be evaluated because of pile up rejection.																														
10000011	An event was triggered, but could not be evaluated by the jitter correction algorithm.																														
10000100	An event was triggered, but could not be evaluated because of a subsequent event.																														
10000101	Beginning of an ADC overflow or underflow.																														
10000110	End of an ADC overflow or underflow <sup>9</sup> .																														
10000111	Beginning of a discarded cycle <sup>10</sup> .																														
10001000	Stop of the measurement because of reaching the preset real time.																														
1 byte:	XXXXXXXX	0 ... 191																													
2 byte:	11XXXXXXXX XXXXXXXX	192 ... 12,479																													
3 byte:	1111XXXX XXXXXXXX XXXXXXXX	12,480 ... 798,911																													
4 byte:	111111XX XXXXXXXX XXXXXXXX XXXXXXXX	798,912 ... 67,907,775																													

8 The list mode 4 uses a 16k spectrum regardless of the setting at CMD\_QUERY\_STATE, byte offset 36.  
9 At the start of the measurement, the MCA527 firmware assumes an ADC overflow for the time before. Therefore, the list always begins with an 'end of ADC overflow' entry.  
10 If the processor is overloaded, it discards cycles of 4000 time units (= 4000 \* 100 nanoseconds = 40 microseconds).

<b>List entries (Continuation)</b>	<p><b>Time coding method 1:</b> Each unsigned int8 value represents the number of time units which are bygone since the previous event. The time unit length is defined in the basis file block. Time intervals that exceeds the value of 255 are coded as non-event information. 11XXXXXX means that no event has occurred for <math>(X+1) * 256</math> time units.</p> <p><b>Time coding method 2:</b> Each unsigned int16 value represents the number of time units which are bygone since the previous event. The time unit length is defined in the basis file block. Time intervals that exceeds the value of 65535 are coded as non-event information. 11XXXXXX means that no event has occurred for <math>(X+1) * 65536</math> time units.</p>
<b>RS232 received data</b>	This file block exists if the used memory size (basis file block, byte offset 72) is greater than zero <sup>11</sup> , and 'Extension port part A configuration' (basis file block, byte offset 102) is 'RS232 (TX/RX, with data buffering)' (=5) or 'Extension port part C configuration' (basis file block, byte offset 104) is 'RS232 (RX, with data buffering)' (=5). The file block is always 1024 bytes long.
<b>Remarks</b>	This file block is optional. It is a freely defined file block.

<sup>11</sup> If the used memory size is equal to zero, the file contains no measurement data, and consequently also no RS232 received data.