

MEA2100 User Guide

Aus Multi Channel Systems Wiki

Inhaltsverzeichnis

- 1 Introduction
 - 1.1 System Overview
 - 1.2 Getting Started
 - 1.3 DSP
 - 1.3.1 Data Acquisition
 - 1.3.1.1 General Overview
 - 1.3.1.2 DSP Reception of Sweep and Digital data
 - 1.3.2 Stimulation
 - 1.3.2.1 General Overview
 - 1.3.2.2 Stimulus Pattern Upload
 - 1.3.2.2.1 Direct Units
 - 1.3.2.2.2 Sideband Data
 - 1.3.2.3 Electrical Configuration
 - 1.3.2.3.1 Stimulus setup advice
 - 1.3.2.3.2 Stimulus channel setup advice
 - 1.3.2.3.3 List mode for advanced simulation
 - 1.3.2.4 Trigger Setup
 - 1.3.2.4.1 Standard Trigger Setup
 - 1.3.2.4.2 Start Trigger
 - 1.3.2.4.2.1 Single mode triggered by SW
 - 1.3.2.4.2.2 Direct Streaming Trigger Setup
 - 1.3.2.5 Direct Streaming Data Path Setup
 - 1.3.3 Power
 - 1.3.3.1 Using MeCalab/pedate to review configuration
 - 1.3.4 Register Interface - 16 bit
 - 2 Data Stream - 16 bit
 - 3 Register Interface

Introduction**System Overview**

The MEA2100 system consists of an interface board and one or two headstages.

The interface board consists of:

- Two USB connectors for PC connection
- Two ports for one or two headstages
- 16 input and 16 output pins for triggering
- Eight additional analog inputs for monitoring
- DSP debug port
- Analog output
- A DSP
 - Who has access to the measured data stream for additional prompt data calculation
 - Optional Stimulation control

The DSP can be programmed by the user, for example to do real-time signal analyses and program feedback via the stimulation channels. A debug port is provided at the same connector as the analog inputs.

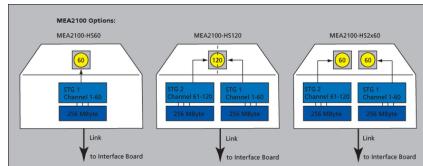


The headstage consists of:

- Adapters for different MEAs (see options below)
- Electronic for measurement
- Stimulation of up to 120 electrodes

The headstages connect to an interface board.

The overview about HS options:

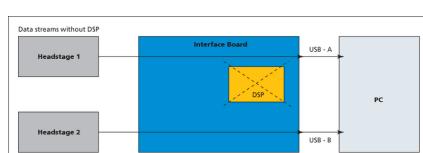
**Getting Started**

To write code for the DSP, the following tools are needed:

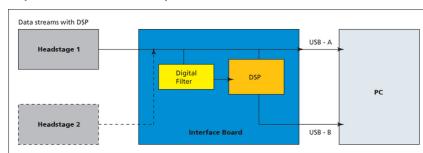
- Code Composer Studio from Texas Instruments, which is the C compiler for the DSP [1] (http://processors.wiki.ti.com/index.php?Category:Code_Composer_Studio_v4)
- Documentation for this project [2] (<http://www.ni.com/tg/ta/ug1870/spru170.pdf>)
- 64bit default package c:\ti\cc\v15.1\TMS320C6454\pack\
- ActivePython [3] (<http://www.activestate.com/activepython/downloads>)
- Visual Studio 2010, there is a free Express Edition available from Microsoft [4] (<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-studio-express>)
- A Blackhawk XDS560v2 System Trace Emulator is of great help to debug DSP code [5] (<http://www.blackhawk-dsp.com/products/XDS560v2.aspx>). The part number is BH-XDS-560v2-BP. A reseller for US is <http://www.corelis.com> (http://www.corelis.com/products/blackhawk/XDS560_Trace_System.htm)

DSP

The default setup of the system routes the data without DSP interaction from HS 1 to USB A and from HS2 to USB B Port. In this case the DSP is only able to monitor the sweep data.



The DSP can be programmed with user code to access the data stream from the headstages and the onboard ADCs of the interface board. The datastream can be send to the USB ports towards the PC. The DSP can also configure all parts of the MEA2100-System, including stimulation and trigger, except the USB data stream configuration to the PC. By this a realtime feedback with low latency can be build.

The DSP used in the MEA2100 system is a TMS320C6454 (<http://www.ti.com/product/tms320c6454>) from Texas Instruments running at 819.2 MHz. He has access to the MEA2100 system via its external memory interface (EMIF). Access to the system is via three parts.

- The Control of the MEA2100 via its own memory based interface. Therefore the DSP has the MEA addresses mapped into the range from 0xA0000000 to 0xBFFFFFFF. The different components of the MEA2100 use each a subsection of this address region #Register Map in the FB_Example Project. #ME2100.h defines Macros which can be used to access these registers:

```

#define ME2100_HREGISTERS(rw) #define ME2100_HREGISTERS(rw)
#define ME2100_HREGISTERS(rw) #define ME2100_HREGISTERS(rw)
#define ME2100_HREGISTERS(rw) #define ME2100_HREGISTERS(rw)
  
```

- Sweep and Digital data is transferred via a fifo-type interface, mapped into the DSP address range starting from 0xB0000000.

- One of two options of controlling the stimulation is the direct streaming mode. This can be done via the streaming port using address 0xC0000000.

Data Acquisition**General Overview**

The MEA2100 System can measure analog signals from 120 electrodes per Headstage at 50kHz with 24bit resolution.

On the interface board are 8 additional analog inputs for monitoring purposes at 50kHz and 24bit resolution.

Additional to the analog signals are several digital signals/configurable DSP in data.

DSP Reception of Sweep and Digital data

Sweep and Digital Data is available to the DSP through a FIFO type interface mapped to the address 0xB0000000 in the DSP memory map.

Each time (once every 20 ms), when the MEA2100 has a new sweep of data available from the Datasources, the signal line (GP04) to the DSP will be toggled (enabled in Interconnection Logic). After the DSP has received this signal, it can read the new sweep data either by multiple reads from address 0xB0000000, or the more convenient way is to set up a DMA transfer which automatically transfers all data from the sweep into DSP memory. After the transfer is complete, a DMA completion interrupt can be called.

The example code in the FB_Example project initializes and enables such a DMA transfer. In the example, each time after a new sweep of MEA2100 is available in DSP memory, the function #InterceptGP04() is called. The new data is available to the user in the array #MeaData[]. The data in this #MeaData[] array automatically updates CPU intervention no loss of CPU time in this mode.

One important Register for the readout of data is Register 0x400. This register controls which kind of data is transferred into the FIFO from MEA2100 to the DSP. It can enable data from Headstage 1, Headstage 2, or from the ADCs on the Interface board and some other helpful data.

The line # of main.c

```

#define MEA2100_MEASURED_CHANNELS 120
#define MEA2100_MEASURED_CHANNELS 120
#define MEA2100_MEASURED_CHANNELS 120
  
```

enables the data transfer from the FPGA to the FIFO towards the DSP.

The ADC Data blocks have the format

```

#def MEA2100_ADC_BLOCK_SIZE 120
#def MEA2100_ADC_BLOCK_SIZE 120
#def MEA2100_ADC_BLOCK_SIZE 120
  
```


THE ASSIGNMENT OF THE THREE LOCAL PATHS AND THREE SIDEBANDS TO THE THREE TRIGGERS WHICH EACH BLOCK IS UNDERRUN IF EXPRESSIONS MATCH AND THE LOCAL AND THE EXPRESSION MATCHES FOR THE SIDEBOARD CHANNELS. WITH THIS SETUP THE CONTROLLING TRIGGER FOR EACH DAC AND SIDEBOARD HAS TO BE CHOSEN.

Start and Stop of the stimulation is always controlled at the level of the triggers, so that all DAC and sidebands which are grouped together to a specific trigger are started and kept running in sync.

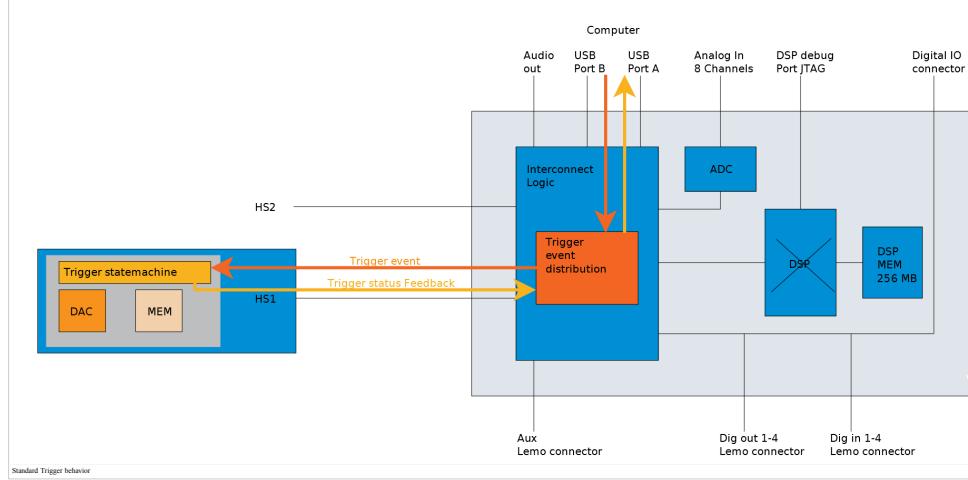
Start Trigger

For the trigger to work, the first bit(0) in the Trigger Control Register 1 (0x200) on the Interface board has to be set to 1. This enables the trigger event packet to propagate from the interface board to the stimulus generator.

The Trigger ID registers (0x210 to 0x214) define for each trigger, which Stimulus-Pattern Memory-Segment to use when the corresponding trigger is started. For single segment mode, the value register at the default value is 0.

To actually start a trigger manually, write a "1" to the bit which corresponds the trigger number in the Trigger Event Status register 0x214. For example to manually start trigger 1, write a value of 0x00000001 to register 0x214.

To start a trigger with external signals or other sources a Digital Multiplexer for source select is implemented (0x280-0x2AC).



Direct Streaming Mode triggered by SW

A further advanced mode for Stimulation is the DSP generated stimulus pattern direct streaming mode, which can be standard triggered and monitored by the computer SW. In this mode the trigger statemachine and memory on an STG will be bypassed and the DAC and SBS data will be send direct from the DSP to the stimulus logic. The setup for stimulus switch and selector still needs to be done as in standard mode.

Direct Streaming Trigger Setup

To detect a trigger event on the DSP the digital data stream to the DSP needs to be enabled reg 0x400 bit 12. In this digital data stream the 8th and 18th vector reflect the trigger event information.

Direct Streaming Data path Setup

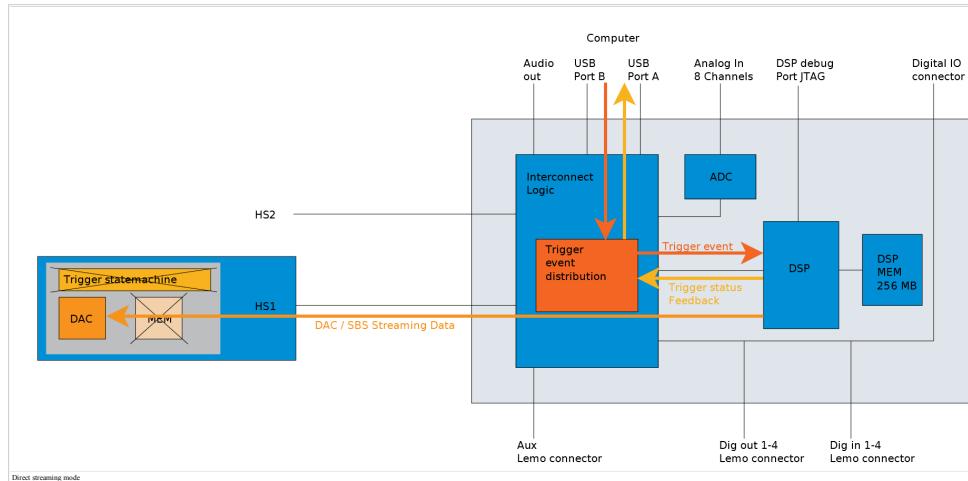
For controlling the stimulation by direct streaming data to an STG, the data source for the DAC and SBS Data need to be switched to DSP. This needs to be done in register 0x400.

To start the distribution of the streaming data from the FIFO configure the register 0x430. After this the stimulus data can be send via the streaming FIFO of the DSP.

For this write 36 DWords to the address 0x20000000 from the DSP:

- 24 DWords for DAC Data
 - 6 DWords to HS1 STG1 (DAC A, C, E, B, D, F)
 - 6 DWords to HS1 STG2 (DAC A, C, E, B, D, F)
 - 6 DWords to HS2 STG1 (DAC A, C, E, B, D, F)
 - 6 DWords to HS2 STG2 (DAC A, C, E, B, D, F)
- 12 DWords for SBS Data
 - 3 DWords to HS1 STG1 (SBS 1, 2, 3)
 - 3 DWords to HS1 STG2 (SBS 1, 2, 3)
 - 3 DWords to HS2 STG1 (SBS 1, 2, 3)
 - 3 DWords to HS2 STG2 (SBS 1, 2, 3)

Always write 36 DWords, even when you switch only one STG to direct streaming mode, because the FIFO read logic expects 36 Data in the predefined order.



Digital Multiplexer

There are multiple digital signals within the MEA2100 system, which represent internal states of the systems. These digital signals can be mapped

- To the digital outputs of the Interface board
- Into the digital datastream via the USB connection to the PC
- Using the digital inputs of the Interface board
- With some restrictions used as triggers conditions for Stimulation
- With some restrictions used as list mode ID increment
- Trigger gated data start mode

The following digital signals are available for Digital Output and Digital Stream to the computer and DSP

- Digital bit 0 to 31; these are 32 bit taken from the rear side Digital Connector
- Digital Pulse Register bit 0 to 31; taken from Digital Pulse Register 0x704-0x710, started by a write to Register 0x700
- Feedback Register bit 0 to 31; taken from Feedback Register 0x700
- Aux_In bit 0 and 1; taken from the two lemo connectors on the Interface board.
- A fixed value of "0"
- A fixed value of "1"
- Trigger Status of the Stimulation Boards
- Any bit from all the Sideband Channels

There is a multiplexer for each bit of the digital datastream at register 0x880-0x8BC for USB connector A and register 0x8C0-0x8FC for USB connector B. For the digital output there is a bit for a register at 0x840-0x87C. There you can select its source from all selectable sources as described in the table above.

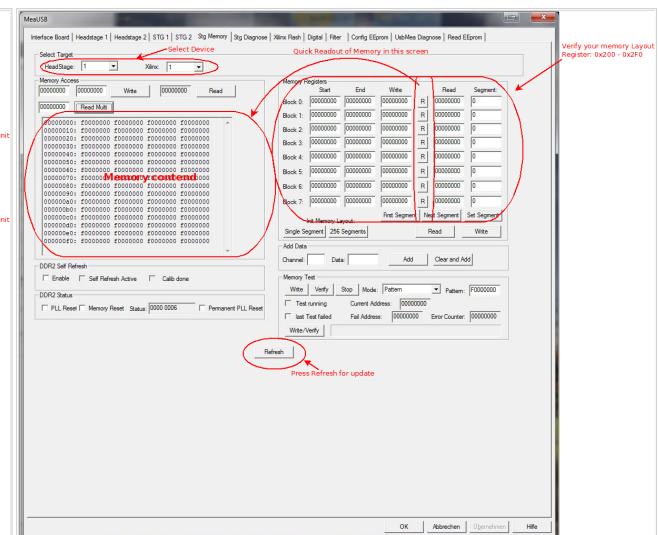
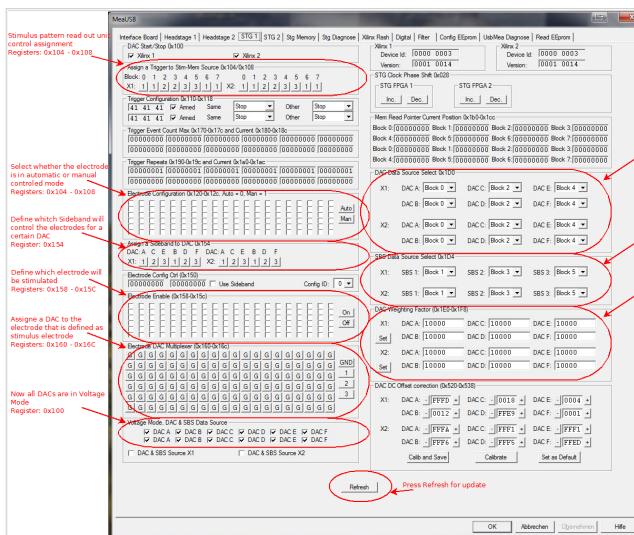
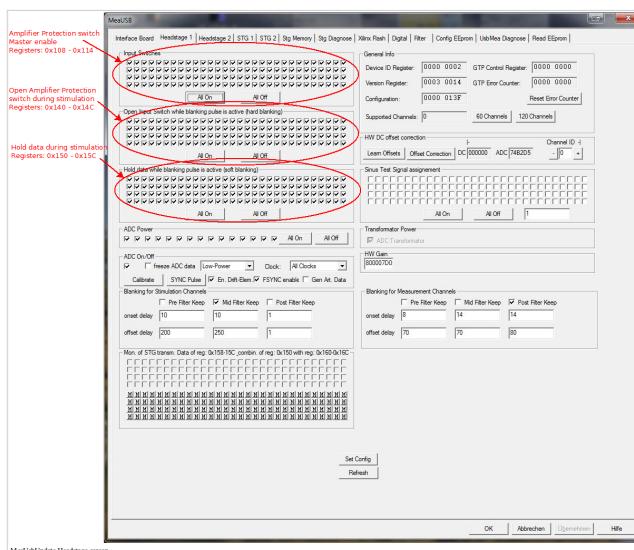
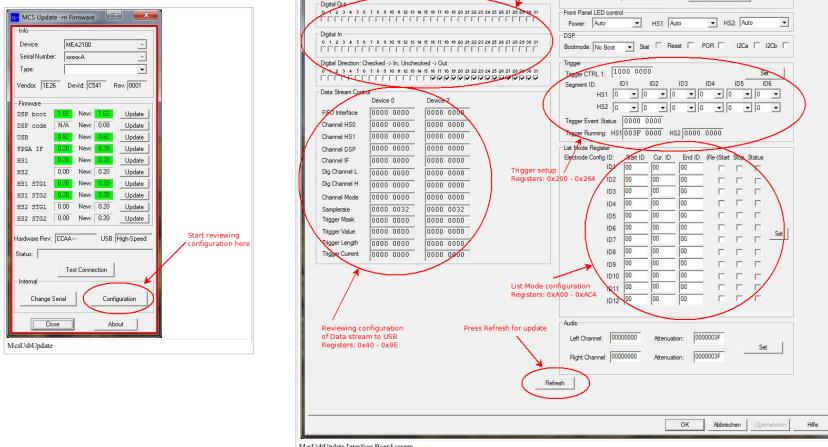
The multiplexers for the following digital signals are available to trigger the Stimulators

- A fixed value of "0"
- Digital bit 0 to 31; these are 32 bit taken from the rear side Digital Connector
- Feedback Register bit 0 to 31; taken from Feedback Register 0x700
- Aux_In bit 0 and 1; taken from the two lemo connectors on the Interface board.

The multiplexers for each trigger are located in register 0x280 to 0x2AC.

Using MstUsbUpdate to review configuration

For reviewing the setup done with the DSP there is a program called MstUsbUpdate. With this program all relevant registers are reflected and according to the use case RO, RW, WO.



Power and Reset Register (should be implemented in every Device)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x020: FPGA Reset	Reserved																													FPGA Reset			
0x024: Power Enable	Reserved																														HS2 PWR Enable	HS1 PWR Enable	IF Trafo
0x028: Trafo startup delay	Reserved																														IF Delay		
0x02C: LED config Register	Reserved																														LED HS 1/2 and IF register mode enable	Reserved	LED HS 1/2 and IF output register

Flash Memory Registers (should be implemented in every Device)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x030: (W)O/J Flash Instruction Code Register	Reserved																														Stop instruction	Fifo Reset	Instruction Code	
0x030: (R)O/J Flash Status Register	Reserved																														FIFO_empty	FIFO_full	Statemachine busy	Flash Status Register
0x034: Flash Memory Address Register	Reserved																														Flash Address			
0x038: Flash Data FIFO Register	256 Data Bytes in 64 DWords to from Flash																																	
0x03C: Flash Clock Divider Register	Multiples of 2 divide 38.4 MHz																																	

Data Stream CTRL Registers

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
0x040: FIFO Interface CTRL Dev0	Reserved																														inl. EOF	inl. Timestamp	inl. Packet cnt.	Reserved	inl. SOF				
0x042: FIFO Interface CTRL Dev2	Reserved																														inl. EOF	inl. Timestamp	inl. Packet cnt.	Reserved	inl. SOF				
0x044: (W)Set FIFO Interface CTRL Dev0	Reserved																														inl. EOF	inl. Timestamp	inl. Packet cnt.	Reserved	inl. SOF				
0x046: (W)Set FIFO Interface CTRL Dev2	Reserved																														inl. EOF	inl. Timestamp	inl. Packet cnt.	Reserved	inl. SOF				
0x048: (W)Clear FIFO Interface CTRL Dev0	Reserved																														inl. EOF	inl. Timestamp	inl. Packet cnt.	Reserved	inl. SOF				
0x04A: (W)Clear FIFO Interface CTRL Dev2	Reserved																														inl. EOF	inl. Timestamp	inl. Packet cnt.	Reserved	inl. SOF				
0x04C: (R)OJEnabled Analog Channels Dev0	Reserved																																		IF	HS2	HS1		
0x04E: (R)OJEnabled Analog Channels Dev2	Reserved																																		IF	HS2	HS1		
0x050: (R)OJEnabled Digital Channels Dev0	Reserved																																		IF	HS2	HS1		
0x052: (R)OJEnabled Digital Channels Dev2	Reserved																																		IF	HS2	HS1		
0x054: Channel config HS1 Dev0	Reserved																														ChannelOffset	Reserved	# of Channels						
0x056: Channel config HS1 Dev2	Reserved																														ChannelOffset	Reserved	# of Channels						
0x058: Channel config HS2 Dev0	Reserved																														ChannelOffset	Reserved	# of Channels						
0x05A: Channel config HS2 Dev2	Reserved																														ChannelOffset	Reserved	# of Channels						
0x05C: Channel config DSP Dev0	Reserved																														ChannelOffset	Reserved	# of Channels						
0x05E: Channel config DSP Dev2	Reserved																														ChannelOffset	Reserved	# of Channels						
0x060: Channel config IF Dev0	Reserved																														ChannelOffset	Reserved	# of Channels						
0x062: Channel config IF Dev2	Reserved																														ChannelOffset	Reserved	# of Channels						
0x064 - 0x06F: Reserved	Reserved																																						
0x070: Digital Data 1 low half Enable Dev0	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x072: Digital Data 1 low half Enable Dev2	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x074: Digital Data 1 up half Enable Dev0	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x076: Digital Data 1 up half Enable Dev2	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x080: Digital Data 2 low half Enable Dev0	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x082: Digital Data 2 low half Enable Dev2	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x084: Digital Data 2 up half Enable Dev0	Reserved																														HS2	HS2	HS2	HS2	HS2				
0x088: FIFO/Channel2 Mode CTRL Dev0	Reserved																														16bit, 24bit, 32 bit mode CTRL.*	Reserved	Cy distinction	Reserved	Triggered Mode				
0x08A: FIFO/Channel Mode CTRL Dev2	Reserved																														16bit, 24bit, 32 bit mode CTRL.*	Reserved	Cy distinction	Reserved	Triggered Mode				
0x08C: Sampling Freq Dev0	Reserved																																						
0x08E: Sampling Freq Dev2	Reserved																																						
0x090: Gate Mask Dev0	Reserved																																						
0x092: Gate Mask Dev2	Reserved																																						
0x094: Compare Value of Gate Mask Dev0	Compare this Value against the Multiplexed Data configured in Register 0x900																																						
0x096: Compare Value of Gate Mask Dev2	Compare this Value against the Multiplexed Data configured in Register 0x900																																						
0x098: Amount of sweeps in Triggered Mode Dev0	Count Start Value																																						
0x09A: Amount of sweeps in Triggered Mode Dev2	Count Start Value																																						
0x09C: Current Count Value Dev0	Current remaining sweeps until Stop																																						
0x0B0: Endpoint FIFO Reset	Reserved																																						
0x0F0: I2C Blocking for other Cypress	Reserved																																						

Decoding Table:

Description:	
0x0000	bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0x0001	bits 23 & 15 aligned
0x0002	bits

Page 1

```

HSD 122 Datwords (Header, 128 Data, Counter)
HSD 123 Datwords (Header, 128 Data, Counter)
HSD 124 Datwords (Header, 128 Data, Counter)
Filtered HSD 122 Datwords (Header, 128 Data, Counter)
Filtered HSD 123 Datwords (Header, 128 Data, Counter)
Filtered HSD 124 Datwords (Header, 128 Data, Counter)
Digital 32 Datwords (Header, 31x Digital Data)
Digital 32 Datwords (Header, 31x Digital Data)
- Digital Max Data for 0x8-10
  - Digital 1 (32 bits)
  - Digital 2 (32 bits)
  - Digital 3 (32 bits)
  - Digital 4 (32 bits)
  - Digital Register Data (sub8)
    - Digital 1 (32 bits)
    - Digital 2 (32 bits)
    - Digital 3 (32 bits)
    - Digital 4 (32 bits)
- Aux 32:
  - Aux 1: Input Level of Aux 1
  - Aux 2: Input Level of Aux 2
- Trigger 8 + 11 Trigger Status (8 bits per trigger: 00: not Armed, 01: Armed, 10: Triggered (running), 11: Reserved)
  Bits 0 + 11 Trigger Status (8 bits per trigger: 00: not Armed, 01: Armed, 10: Triggered (running), 11: Reserved)
  Bits 0 + 11 Trigger Status (8 bits per trigger: 00: not Armed, 01: Armed, 10: Triggered (running), 11: Reserved)
  Bits 24 + 29 Trigger Event Type (1 trigger event occurs)
  Bits 24 + 29 Trigger Event Type (1 trigger event occurs)
  Bits 0 + 11 Headband 1: Sideband 1 From STG 1
  Bits 12 + 23 Headband 1: Sideband 2 From STG 1
  Bits 0 + 11 Headband 1: Sideband 1 From STG 2
  Bits 12 + 23 Headband 1: Sideband 2 From STG 2
  Bits 0 + 11 Headband 1: Sideband 1 From STG 1
  Bits 12 + 23 Headband 1: Sideband 2 From STG 1
  Bits 0 + 11 Headband 1: Sideband 1 From STG 2
  Bits 12 + 23 Headband 1: Sideband 2 From STG 2
  - DAC 1 From HSD of STG1:
    Bits 0 + 11 Headband 1: DAC A From STG 1
    Bits 12 + 23 Headband 1: DAC B From STG 1
    Bits 0 + 11 Headband 1: DAC C From STG 1
    Bits 12 + 23 Headband 1: DAC D From STG 1
  - DAC 1 From HSD of STG2:
    Bits 0 + 11 Headband 1: DAC E From STG 1
    Bits 12 + 23 Headband 1: DAC F From STG 1
    Bits 0 + 11 Headband 1: DAC G From STG 1
    Bits 12 + 23 Headband 1: DAC H From STG 1
  - DAC 2 From HSD of STG2:
    Bits 0 + 11 Headband 1: DAC I From STG 2
    Bits 12 + 23 Headband 1: DAC J From STG 2
    Bits 0 + 11 Headband 1: DAC K From STG 2
    Bits 12 + 23 Headband 1: DAC L From STG 2
  - Seg_Alice_0: 0x from HSD of STG1:
    Bits 0 + 11: current Segment ID
    Bits 12 + 23: current Segment ID
    Bits 0 + 11: current Segment ID
    Bits 12 + 23: current Segment ID
  - Seg_Alice_1: 0x from HSD of STG1:
    Bits 0 + 11: current Segment ID
    Bits 12 + 23: current Segment ID
    Bits 0 + 11: current Segment ID
    Bits 12 + 23: current Segment ID

```

```
The header has the format  
Bit 31 : 1, when data from a headstage is enabled and the HS is not connected, otherwise 0  
Bit 24 to 30 : Data Source enumeration (see below)  
Bit 9 to 23 : Reserved (Always Zero)  
Bit 8 to 7 : The number of datapoints + counter values following this header, 0x79 for Headstage data (120 +
```

```
Data Source enumeration:  
1: Headstage 1  
2: Headstage 2  
3: Analog Data from Inter  
4: Headstage 1 filtered I  
5: Headstage 2 filtered I  
6: Digital Data  
7: Timestamp counter
```

Audio DAC Register

```
Source decoding:  
0: No source  
1: HSC1  
2: HSC2  
3: IF  
4: DSP(bits 23 - 0)
```

Digital Data Register

*Decoding Table

```

Digital In 0 Detach (bit 2)
Digital In 1 Attach (bit 3)
Digital In 2 (bit 4)
Aux In (bit 5)
Digital Out 0 Status from HSC (bit 6)
SBS From HSC Channel 1 or X1 and X2 (bit 8)
SBS From HSC Channel 2 or X3 and X4 (bit 9)
SBS From HSC Channel 1 or X1 and X2 (bit 11)
SBS From HSC Channel 2 or X3 and X4 (bit 12)
DAC C-0 From HSC #1 (bit 13)
DAC C-0 From HSC #2 (bit 14)
DAC E-0 From HSC #1 (bit 15)
DAC E-0 From HSC #2 (bit 16)
DAC E-0 From HSC #3 (bit 17)
DAC E-0 From HSC #4 (bit 18)
Trigger Statechange Status from HSC (bit 28)
SBS From HSC Channel 1 or X1 and X2 (bit 29)
SBS From HSC Channel 2 or X3 and X4 (bit 30)
SBS From HSC Channel 1 or X1 and X2 (bit 31)
SBS From HSC Channel 2 or X3 and X4 (bit 32)
DAC C-0 From HSC #1 (bit 33)
DAC C-0 From HSC #2 (bit 34)
DAC E-0 From HSC #1 (bit 35)
DAC E-0 From HSC #2 (bit 36)
DAC E-0 From HSC #3 (bit 37)
DAC E-0 From HSC #4 (bit 38)

```

List Mode Register

Mini DMA Registers (only on Rev with USB 3.0)

Register

Register	Description	Reset Value	Access	Fifo Reset	Instruction Code
0xF00 (WO)EEPROM Instruction Code Register	Reserved				
0xF00 (R)EEPROM Status Register	Reserved			FIFO_empty, FIFO_full Statemachine busy	Flash Status Register
0xF04 EEPROM Memory Address Register	Reserved		Flash Address		
0xF08 EEPROM Data FIFO Register	256 Data Bytes in 64 WordBlocks from Flash				
0xF0C : EEPROM HW configuration Register	Address length (1..2 Bytes), Clock Divider Register (Multiples of 2 divide 38.4 MHz)				
0xF10 : EEPROM Offset Register	Offset for reads and writes to the Eeprom				
0xF14 : EEPROM Size Register	Size of the Eeprom Block available for this Image				

Multiboot Registers (only on Rev with USB 3.0)

卷之三十一

Register	from	to	in Revision	USB Rev
Electrode config/segment ID register	0x0208 .. 0x0234	0x021B .. 0x0244	0.05	0.27
Event Status Register	0x0238	0x0244	0.05	0.27
ADC Filter	0x0120 .. 0x0138	0x0060 .. 0x0074	0.05	
FIFO Channel Model CTRL Dex/X	0x0068 .. 0x006B	0x006C .. 0x006F	0.05	
FIFO Channel Free Dex/X	0x006C .. 0x006F	0x0070 .. 0x0073	0.05	
DIGITAL_CHANNELS_L	0x0064 .. 0x0067	0x0080 .. 0x0083	0.07	0.42
DIGITAL_CHANNELS_H	0x0068 .. 0x006B	0x0084 .. 0x0087	0.07	0.42
FIFO Channel Model CTRL Dex/X	0x006C .. 0x006F	0x0088 .. 0x008B	0.07	0.42
FIFO Channel Free Dex/X	0x0070 .. 0x0073	0x008E .. 0x00BF	0.07	0.42
Select menu for Dex Data	0x0340 .. 0x034D	0x0840 .. 0x0849	0.09 .. 0.51	

```
*** Decoding Table:
-----[value]-----[Source]
000 : '0' (default for all bits)
001 : 1 Digital In bit 11 downto 0
002 : Feedback bit 11 downto 0
003 : 1 Digital In bit 10 downto 0
004 : 1 Digital In bit 9 downto 0
005 : 1 Digital In bit 8 downto 0
006 : 1 Digital Pulse Register bit 11 downto 0
007 : 1 Digital Pulse Register bit 10 downto 0
008 : 1 Digital Pulse Register bit 9 downto 0
009 : 1 Digital Pulse Register bit 8 downto 0
010 : 1 Digital In bit 7 downto 0
011 : 1 Digital In bit 6 downto 0
012 : 1 Digital In bit 5 downto 0
013 : 1 Digital In bit 4 downto 0
014 : 1 Digital In bit 3 downto 0
015 : 1 Digital In bit 2 downto 0
016 : 1 Digital In bit 1 downto 0
017 : 1 Digital In bit 0 downto 0
018 : DAQ Cypress 1 Virtual Device 1 is started
019 : DAQ Cypress 1 Virtual Device 2 is started
020 : DAQ Cypress 1 Virtual Device 3 is started
021 : DAQ Cypress 1 Virtual Device 4 is started
022 : DAQ Cypress 1 Virtual Device 5 is started
023 : DAQ Cypress 1 Virtual Device 6 is started
024 : DAQ Cypress 1 Virtual Device 7 is started
025 : DAQ Cypress 1 Virtual Device 8 is started
026 : DAQ Cypress 1 Virtual Device 9 is started
027 : DAQ Cypress 1 Virtual Device 10 is started
028 : DAQ Cypress 1 Virtual Device 11 is started
029 : DAQ Cypress 1 Virtual Device 12 is started
030 : DAQ Cypress 1 Virtual Device 13 is started
031 : DAQ Cypress 1 Virtual Device 14 is started
032 : DAQ Cypress 1 Virtual Device 15 is started
033 : DAQ Cypress 1 Virtual Device 16 is started
034 : DAQ Cypress 1 Virtual Device 17 is started
035 : DAQ Cypress 1 Virtual Device 18 is started
036 : DAQ Cypress 1 Virtual Device 19 is started
037 : DAQ Cypress 1 Virtual Device 20 is started
038 : DAQ Cypress 1 Virtual Device 21 is started
039 : DAQ Cypress 1 Virtual Device 22 is started
040 : DAQ Cypress 1 Virtual Device 23 is started
041 : DAQ Cypress 1 Virtual Device 24 is started
042 : DAQ Cypress 1 Virtual Device 25 is started
043 : DAQ Cypress 1 Virtual Device 26 is started
044 : DAQ Cypress 1 Virtual Device 27 is started
045 : DAQ Cypress 1 Virtual Device 28 is started
046 : DAQ Cypress 1 Virtual Device 29 is started
047 : DAQ Cypress 1 Virtual Device 30 is started
048 : DAQ Cypress 1 Virtual Device 31 is started
049 : DAQ Cypress 1 Virtual Device 32 is started
050 : DAQ Cypress 1 Virtual Device 33 is started
051 : DAQ Cypress 1 Virtual Device 34 is started
052 : DAQ Cypress 1 Virtual Device 35 is started
053 : DAQ Cypress 1 Virtual Device 36 is started
054 : DAQ Cypress 1 Virtual Device 37 is started
055 : DAQ Cypress 1 Virtual Device 38 is started
056 : DAQ Cypress 1 Virtual Device 39 is started
057 : DAQ Cypress 1 Virtual Device 40 is started
058 : DAQ Cypress 1 Virtual Device 41 is started
059 : DAQ Cypress 1 Virtual Device 42 is started
060 : DAQ Cypress 1 Virtual Device 43 is started
061 : DAQ Cypress 1 Virtual Device 44 is started
062 : DAQ Cypress 1 Virtual Device 45 is started
063 : DAQ Cypress 1 Virtual Device 46 is started
064 : DAQ Cypress 1 Virtual Device 47 is started
065 : DAQ Cypress 1 Virtual Device 48 is started
066 : DAQ Cypress 1 Virtual Device 49 is started
067 : DAQ Cypress 1 Virtual Device 50 is started
068 : DAQ Cypress 1 Virtual Device 51 is started
069 : DAQ Cypress 1 Virtual Device 52 is started
070 : DAQ Cypress 1 Virtual Device 53 is started
071 : DAQ Cypress 1 Virtual Device 54 is started
072 : DAQ Cypress 1 Virtual Device 55 is started
073 : DAQ Cypress 1 Virtual Device 56 is started
074 : DAQ Cypress 1 Virtual Device 57 is started
075 : DAQ Cypress 1 Virtual Device 58 is started
076 : DAQ Cypress 1 Virtual Device 59 is started
077 : DAQ Cypress 1 Virtual Device 60 is started
078 : DAQ Cypress 1 Virtual Device 61 is started
079 : DAQ Cypress 1 Virtual Device 62 is started
080 : DAQ Cypress 1 Virtual Device 63 is started
081 : DAQ Cypress 1 Virtual Device 64 is started
082 : DAQ Cypress 1 Virtual Device 65 is started
083 : DAQ Cypress 1 Virtual Device 66 is started
084 : DAQ Cypress 1 Virtual Device 67 is started
085 : DAQ Cypress 1 Virtual Device 68 is started
086 : DAQ Cypress 1 Virtual Device 69 is started
087 : DAQ Cypress 1 Virtual Device 70 is started
088 : DAQ Cypress 1 Virtual Device 71 is started
089 : DAQ Cypress 1 Virtual Device 72 is started
090 : DAQ Cypress 1 Virtual Device 73 is started
091 : DAQ Cypress 1 Virtual Device 74 is started
092 : DAQ Cypress 1 Virtual Device 75 is started
093 : DAQ Cypress 1 Virtual Device 76 is started
094 : DAQ Cypress 1 Virtual Device 77 is started
095 : DAQ Cypress 1 Virtual Device 78 is started
096 : DAQ Cypress 1 Virtual Device 79 is started
097 : DAQ Cypress 1 Virtual Device 80 is started
098 : DAQ Cypress 1 Virtual Device 81 is started
099 : DAQ Cypress 1 Virtual Device 82 is started
```

Mailbox Register Address Map (Address bits 11-0) Base Address: 0x1000

Mailbox to DSP	
Register	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0x000 - 0xFFC: Mailbox Registers	write will generate an Interrupt on GPIO 6 Line

RAM Register Address Map (Address bits 11-0) Base Address: 0x2000:

Headstage Board Address Map (Address bits 11-0) Base Address: HS1: 0x8000 HS2: 0xC000

Access time is 2.2 us for writes and 2.4 us for reads

General Purpose Registers (should be implemented in every Device)

***Decoding Table:**

```
#: 2 x 60 MCA  
#: 1 x 60 MCA  
#: 1 x 120 MCA  
#: special
```

.....

GTR Registers (should be implemented in every Device)																																
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0..16_GTR_GTRn_D[31..0]	D ₃₁	D ₃₀	D ₂₉	D ₂₈	D ₂₇	D ₂₆	D ₂₅	D ₂₄	D ₂₃	D ₂₂	D ₂₁	D ₂₀	D ₁₉	D ₁₈	D ₁₇	D ₁₆	D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

0x010: GTP CTRL Register	Reserved
0x014: (R)GTP Error counter Register	Reserved

0x024: Power Enable | Reserved

0x03C : Flash H

0x150	Enable Blanking	Reserved	Electrode 30 - 1
0x154	Enable Blanking	Reserved	Electrode 60 - 31
0x158	Enable Blanking	Reserved	Electrode 90 - 61
0x15C	Enable Blanking	Reserved	Electrode 119 - 91
0x160	Stimulus Data keep pre filter	Reserved	onset delay
0x164	Stimulus Data keep mid filter	Reserved	onset delay
0x168	Stimulus Data keep post filter	Reserved	onset delay
0x170	None Stimulus Data keep mid filter	Reserved	onset delay
0x174	Non-Stimulus Data keep mid filter	Reserved	onset delay
0x178	Non-Stimulus Data keep post filter	Reserved	onset delay
0x180	Channel Index	Reserved	Index used for Reg.: 0x184 - 0x188
0x184	Current ADC Value	Reserved	Reflects the ADC Value of the channel selected in Reg. 0x180
0x188	Current DC Channel offset	Reserved	Reflects the DC offset correction Value of the channel selected in Reg. 0x180
0x1F0	Stimulation Channel copy	Reserved	Electrode 30 - 1 Reflects the STG Register 0x15B and 0x15C for monitoring if information is on HS
0x1F4	Stimulation Channel copy	Reserved	Electrode 60 - 31
0x1F8	Stimulation Channel copy	Reserved	Electrode 90 - 61
0x1FC	Stimulation Channel copy	Reserved	Electrode 119 - 91
0x6000	Filter 1 coefficient a[0]	Filter coefficient b[0] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6040	Reserved (DC)	Reserved	
0x6080	Filter 1 coefficient b[1]	Filter coefficient b[1] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x60C0	Filter 1 coefficient a[1]	Filter coefficient a[1] as Q1.30 value	
0x6100	Filter 1 coefficient b[2]	Filter coefficient b[2] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6140	Filter 1 coefficient a[2]	Filter coefficient a[2] as Q1.30 value	
0x6180	Filter 1 coefficient a[1].ext. and a[2].ext.	Filter Coef. A2 lower bits -31 downto -46	Filter Coef. A1 lower bits -31 downto -46
0x61C0	Filter 1 CTRL	Reserved	Reserved Filter Mode Filter Blank Enable Filter Enable
0x6200	Filter 2 coefficient b[0]	Filter coefficient b[0] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6240	Reserved (DC)	Reserved	
0x6280	Filter 2 coefficient b[1]	Filter coefficient b[1] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x62C0	Filter 2 coefficient a[1]	Filter coefficient a[1] as Q1.30 value	
0x6300	Filter 2 coefficient b[2]	Filter coefficient b[2] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6340	Filter 2 coefficient a[2]	Filter coefficient a[2] as Q1.30 value	
0x6380	Filter 2 coefficient a[1].ext. and a[2].ext.	Filter Coef. A2 lower bits -31 downto -46	Filter Coef. A1 lower bits -31 downto -46
0x63C0	Filter 2 CTRL	Reserved	Reserved Filter Mode Filter Blank Enable Filter Enable
0x6400	Filter 3 coefficient b[0]	Filter coefficient b[0] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6440	Reserved (DC)	Reserved	
0x6480	Filter 3 coefficient b[1]	Filter coefficient b[1] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x64C0	Filter 3 coefficient a[1]	Filter coefficient a[1] as Q1.30 value	
0x6500	Filter 3 coefficient b[2]	Filter coefficient b[2] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6540	Filter 3 coefficient a[2]	Filter coefficient a[2] as Q1.30 value	
0x6580	Filter 3 coefficient a[1].ext. and a[2].ext.	Filter Coef. A2 lower bits -31 downto -46	Filter Coef. A1 lower bits -31 downto -46
0x65C0	Filter 3 CTRL	Reserved	Reserved Filter Mode Filter Blank Enable Filter Enable
0x6600	Filter 4 coefficient b[0]	Filter coefficient b[0] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6640	Reserved (DC)	Reserved	
0x6680	Filter 4 coefficient b[1]	Filter coefficient b[1] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x66C0	Filter 4 coefficient a[1]	Filter coefficient a[1] as Q1.30 value	
0x6700	Filter 4 coefficient b[2]	Filter coefficient b[2] as Q1.16 value	"00000000000000" to extend coefficient to Q1.30
0x6740	Filter 4 coefficient a[2]	Filter coefficient a[2] as Q1.30 value	

0x078: Filter 4 coefficient a[1] least significant bit	Filter Coef. A2 lower bits -31 downto -46																																					
0x07C: Filter 4 CTRL	Reserved		Reserved Mode	Filter Blank Enable	Filter Enable																																	
0x0A0: Filter 2 coefficient b[2][0]	Filter coefficient b2[0] as Q1.16 value for one clock after blank	"0000000000000000" to extend coefficient to Q1.30																																				
0x0A8: Filter 2 coefficient b[2][1]	Filter coefficient b2[1] as Q1.16 value for one clock after blank	"0000000000000000" to extend coefficient to Q1.30																																				
0x0AC: Filter 2 coefficient b[2][1]	Filter coefficient a2[1] as Q1.30 value for one clock after blank																																					
0x0B0: Filter 2 coefficient b[2][2]	Filter coefficient b2[2] as Q1.16 value for one clock after blank	"0000000000000000" to extend coefficient to Q1.30																																				
0x0B4: Filter 2 coefficient a[2][2]	Filter coefficient a2[2] as Q1.30 value for one clock after blank																																					
0x0C0: Filter Info	Corner frequency of Hardware Filter in mHz																																					
0x0E4: Filter Info	Hardware Filter order	Band: Lowpass	Family: Butterworth/Bessel	Reserved	0 = Hardware Filter	Filter Active																																
0x0E0: Filter Info	Corner frequency of Highpass Filter in mHz																																					
0x0D4: Filter Info	Highpass Filter order	Band: Highpass	Family: Butterworth/Bessel	Reserved	1 = Software Filter	Filter Active																																
0x0E6: Filter Info	Corner frequency of Lowpass Filter in mHz																																					
0x0E8: Filter Info	Lowpass Filter order	Band: Lowpass	Family: Butterworth/Bessel	Reserved	1 = Software Filter	Filter Active																																
EEPROM Registers																																						
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x300: (WO)EEPROM Instruction Code Register	Reserved																																					
0x300: (RO)EEPROM Status Register	Reserved																																					
0x304 : EEPROM Memory Address Register	Reserved																																					
0x308 : EEPROM Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																																					
0x30C : EEPROM HW configuration register	Address length (1,2,3 Bytes)	Clock Divider Register (Multiples of 2 divide 38.4 MHz)																																				
Mini DMA Registers																																						
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0xE00: DMA CTRL Register	Reserved																																					
0xE04 : EEPROM Address Ptr Register	Reserved																																					
0xE08 : EEPROM Data Ptr Register	Reserved																																					
Mini DMA command overview																																						
Command	Bit 31 to 24	Bit 23 to 0	Description																																			
STP	0xFF	0XXXXXX	Stop DMA																																			
SDTA	0x01	0x00RegAddr	Store Next Data to Register Address																																			
DATA	0xData	0xData	Data expected after SDTA command																																			
Changed Register Information of Headstage FPGA																																						
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
ADC Filter	0x0120 - 0x0138	0x0690 - 0x0674		0.04																																		
ADC Filter enable	0x100 bits 4:1	0x61C, 0x63C, 0x65C, 0x67C each bit 0	Multiples of 2 divide 38.4 MHz	1.15																																		

Access time is 4.3 us for writes and 6.4 us for reads.																																					
General Purpose Registers (should be implemented in every Device)																																					
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x000: Device ID Register	0x00000001 - MEA 21 Stimulus FPGA																																				
0x04: HW/FPGA Version Register	Reserved																																				
SPI Registers (should be implemented in every Device)																																					
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x10: SPI CTRL Register	Reserved																																				
Reset Register (should be implemented in every Device)																																					
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x20: FPGA Reset (WO)	Reserved																																				
0x24: FPGA Reset Status (R0)	Reserved																																				
0x28: System PH CTRL (WO)	Reserved																																				
Flash Memory Registers (should be implemented in every Device)																																					
Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x030: (W)Flash Instruction Code Register	Reserved																																				
0x030: (R)Flash Status Register	Reserved																																				
0x044: Flash Memory Address Register	Reserved																																				
0x088: Flash Data FIFO Register	256 Data Bytes in 64 DWords to from Flash																																				
0x0C: Flash Clock Divider Register	Multiples of 2 divide 38.4 MHz																																				

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100: DAC Control Register	Reserved																		0: Current I: Voltage DAC: F	0: Current I: Voltage DAC: E	0: Current I: Voltage DAC: D	0: Current I: Voltage DAC: C	0: Current I: Voltage DAC: B	0: Current I: Voltage DAC: A	Reserved	DAC Test	Reserved	DAC data source (0 intern, 1 DSP stream)	Start Stop			
0x104: Trigger select for Stim MEM FSM	Reserved																		Mem 7	Mem 5	Mem 6	Mem 3	Mem 4	Mem 2	Mem 1							
0x108: Trigger select for Stim MEM FSM	Reserved																		Mem 8	Mem 6	Mem 5	Mem 4	Mem 3	Mem 2	Mem 1							
0x10C: Stop Trigger Register	Reserved																												Stop Trigger 3-2-1			
0x110: Trigger count Registers++	Reserved																												Trigger 1			
0x114: Trigger count Registers++	Reserved																												Trigger 2			
0x118: Trigger count Registers++	Reserved																												Trigger 3			
0x11C: Reserved	Reserved																															
0x120: Electrode Mode	Reserved																		Electrode 15 - 1 (Manual mode: 11 / Automatic mode: 00 , other bit combinations are reserved)													
0x124: Electrode Mode	Reserved																		Electrode 30 - 16													
0x128: Electrode Mode	Reserved																		Electrode 45 - 31													
0x12C: Electrode Mode	Reserved																		Electrode 60 - 46													
0x130: Reserved	Reserved																															
0x134: Reserved	Reserved																															
0x138: Reserved	Reserved																															
0x13C: Reserved	Reserved																															
0x140: Reserved	Reserved																															
0x144: Reserved	Reserved																															
0x148: Reserved	Reserved																															
0x14C: Reserved	Reserved																															
0x154: DAC Sideband Select f. Auto	Reserved																		DAC F	Reserved	DAC D	Reserved	DAC B	Reserved		DAC E	Reserved	DAC C	Reserved	DAC A		
0x158: Stimulation Enable	Reserved																		Electr. 30 - 1 (Segment ID depending on bits 7:0 of reg. 0x150)													
0x15C: Stimulation Enable	Reserved																		Electr. 60 - 31 (Segment ID depending on bits 7:0 of reg. 0x150)													
0x160: Electrode DAC select****	Reserved																		Electr. 15 - 1 (Segment ID depending on bits 7:0 of reg. 0x150)													
0x164: Electrode DAC select****	Reserved																		Electr. 30 - 16 (Segment ID depending on bits 7:0 of reg. 0x150)													
0x168: Electrode DAC select****	Reserved																		Electr. 45 - 31 (Segment ID depending on bits 7:0 of reg. 0x150)													
0x16C: Electrode DAC select****	Reserved																		Electr. 60 - 46 (Segment ID depending on bits 7:0 of reg. 0x150)													
0x170: Event Cnt. Max val.	Trigger 1																															
0x174: Event Cnt. Max val.	Trigger 2																															
0x178: Event Cnt. Max val.	Trigger 3																															
0x17C: Reserved	Reserved																															
0x180: Event Cnt. Cur. val.	Trigger 1																															
0x184: Event Cnt. Cur. val.	Trigger 2																															
0x188: Event Cnt. Cur. val.	Trigger 3																															
0x18C: Reserved	Reserved																															
0x190: Repeat Trigger # times	Trigger 1																															
0x194: Repeat Trigger # times	Trigger 2																															
0x198: Repeat Trigger # times	Trigger 3																															
0x19C: Reserved	Reserved																															
0x1A0: Repeat times counter	Trigger 1																															
0x1A4: Repeat times counter	Trigger 2																															
0x1A8: Repeat times counter	Trigger 3																															
0x1AC: Reserved	Reserved																															
0x1B0: Read Pn. Cur. pos.	Stim 1																															
0x1B4: Read Pn. Cur. pos.	Stim 2																															
0x1B8: Read Pn. Cur. pos.	Stim 3																															
0x1BC: Read Pn. Cur. pos.	Stim 4																															
0x1C0: Read Pn. Cur. pos.	Stim 5																															
0x1C4: Read Pn. Cur. pos.	Stim 6																															
0x1C8: Read Pn. Cur. pos.	Stim 7																															
0x1CC: Read Pn. Cur. pos.	Stim 8																															
0x1D0: DAC Data Source select****	Reserved																		DAC F	DAC D	DAC B	Reserved	DAC E		DAC C		DAC A					
0x1D4: SBS Data Source select****	Reserved																												SBS 3	SBS 2	SBS 1	
0x1E0: DAC Weighting Factor	Reserved																															
0x1E4: DAC	n.....																															

Weighting Factor	Reserved	DAC C
0x1E8: DAC Weighting Factor	Reserved	DAC E
0x1F0: DAC Weighting Factor	Reserved	DAC B
0x1F4: DAC Weighting Factor	Reserved	DAC D
0x1F8: DAC Weighting Factor	Reserved	DAC F

SBS: Side Band Signal

*Decoding Table:

00: Trigger 1
01: Trigger 2
10: Trigger 3
11: Reserved

**Decoding Table:

00: Stop stimulus sequence at recording of same trigger event
01: Restart stimulus sequence at recording of same trigger event
10: Stop stimulus sequence at recording of other trigger event
11: Gate stimulus sequence at trigger event

***Decoding Table:

00: SBS 1
01: SBS 2
10: SBS 3
11: Reserved, not valid

****Decoding Table:

0000: Stimulus 1 Data Stream
0001: Stimulus 2 Data Stream
0010: Stimulus 3 Data Stream
0011: Stimulus 4 Data Stream
0100: Stimulus 5 Data Stream
0101: Stimulus 6 Data Stream
0110: Stimulus 7 Data Stream
0111: Stimulus 8 Data Stream

Stimulus Pattern Memory Pointer Registers

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x200: MEM Control Stim 1	Reserved	Init Prt all*	Init Prt Seg.0**	Reserved																												
0x204: Start Pointer Stim 1	Memory Pointer (DAC A and B select this Data source in reg. 0x1D0 by default)																															
0x208: End Pointer Stim 1	Memory Pointer																															
0x20C: Write Pointer Stim 1	Memory Pointer, write will clear Channel																															
0x210: Read Pointer Stim 1	Memory Pointer																															
0x220: MEM Control Stim 2	Reserved																															
0x224: Start Pointer Stim 2	Memory Pointer (SBS 1 select this Data source in reg. 0x1D4 by default)																															
0x228: End Pointer Stim 2	Memory Pointer																															
0x22C: Write Pointer Stim 2	Memory Pointer, write will clear Channel																															
0x230: Read Pointer Stim 2	Memory Pointer																															
0x240: MEM Control Stim 3	Reserved																															
0x244: Start Pointer Stim 3	Memory Pointer (SBS 2 select this Data source in reg. 0x1D4 by default)																															
0x248: End Pointer Stim 3	Memory Pointer																															
0x24C: Write Pointer Stim 3	Memory Pointer, write will clear Channel																															
0x250: Read Pointer Stim 3	Memory Pointer																															
0x260: MEM Control Stim 4	Reserved																															
0x264: Start Pointer Stim 4	Memory Pointer (SBS 2 select this Data source in reg. 0x1D4 by default)																															
0x268: End Pointer Stim 4	Memory Pointer																															
0x26C: Write Pointer Stim 4	Memory Pointer, write will clear Channel																															
0x270: Read Pointer Stim 4	Memory Pointer																															
0x280: MEM Control Stim 5	Reserved																															
0x284: Start Pointer Stim 5	Memory Pointer (DAC E and F select this Data source in reg. 0x1D0 by default)																															
0x288: End Pointer Stim 5	Memory Pointer																															
0x28C: Write Pointer Stim 5	Memory Pointer, write will clear Channel																															
0x290: Read Pointer Stim 5	Memory Pointer																															
0x2A0: MEM Control Stim 6	Reserved																															
0x2A4: Start Pointer Stim 6	Memory Pointer (SHS 3 select this Data source in reg. 0x1D4 by default)																															
0x2A8: End Pointer Stim 6	Memory Pointer																															
0x2AC: Write Pointer Stim 6	Memory Pointer, write will clear Channel																															
0x2B0: Read Pointer Stim 6	Memory Pointer																															
0x2C0: MEM Control Stim 7	Reserved																															
0x2C4: Start Pointer Stim 7	Memory Pointer (unused by default)																															
0x2C8: End Pointer Stim 7	Memory Pointer																															
0x2CC: Write Pointer Stim 7	Memory Pointer, write will clear Channel																															
0x2D0: Read Pointer Stim 7	Memory Pointer																															
0x2E0: MEM Control Stim 8	Reserved																															
0x2E4: Start Pointer Stim 8	Memory Pointer (unused by default)																															
0x2E8: End Pointer Stim 8	Memory Pointer																															
0x2EC: Write Pointer Stim 8	Memory Pointer, write will clear Channel																															
0x2F0: Read Pointer Stim 8	Memory Pointer																															

Segments

Segment 0: 0x0000-0x000f: the Segment ID is the bit 0 of trigger
--

Initialization:

half bit after writing a '1' until bit 14 is '0' to wait an end of request!

EEPROM Registers

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x300: (W)EEPROM Instruction Code Register	Reserved																															
0x300: (R)EEPROM Status Register	Reserved																															
0x304: EEPROM Memory Address Register	Reserved																															
0x308: EEPROM Data FIFO Register	256 Data Bytes in 64 DWords to/from Flash																															
0x30C: EEPROM HW configuration register	Address length (1,2,3 Bytes)																															
0x400: Memory Test Status Register	MEM Test result in process																															
0x404: MEM Test Register	Stop Test																															
0x408: MEM Pattern Register	Pattern																															
0x40C: MEM Test Fail Counter Register	# of data compare fails (writes clear value)																															
0x410: MEM Test Address Register	Reserved</td																															

Register	Byte 4	Byte 3	Byte 2	Byte 1
0x500 MEM Write Address Register	MEM Address(RW)			
0x504 MEM Data Register	Write Data (W0), Read Data(R0)			
0x508 - 0x51C: Reserved	Reserved			
0x520: Write Stim 1 Data Register	Channel 0 Data Vector*			
0x524: Write Stim 2 Data Register	Channel 1 Data Vector*			
0x528: Write Stim 3 Data Register	Channel 2 Data Vector*			
0x52C: Write Stim 4 Data Register	Channel 3 Data Vector*			
0x530: Write Stim 5 Data Register	Channel 4 Data Vector*			
0x534: Write Stim 6 Data Register	Channel 5 Data Vector*			
0x538: Write Stim 7 Data Register	Channel 6 Data Vector*			
0x53C: Write Stim 8 Data Register	Channel 7 Data Vector*			
0x540: Clear and Write Stim 1 Data Register	Channel 0 Data Vector*			
0x544: Clear and Write Stim 2 Data Register	Channel 1 Data Vector*			
0x548: Clear and Write Stim 3 Data Register	Channel 2 Data Vector*			
0x54C: Clear and Write Stim 4 Data Register	Channel 3 Data Vector*			
0x550: Clear and Write Stim 5 Data Register	Channel 4 Data Vector*			
0x554: Clear and Write Stim 6 Data Register	Channel 5 Data Vector*			
0x558: Clear and Write Stim 7 Data Register	Channel 6 Data Vector*			
0x55C: Clear and Write Stim 8 Data Register	Channel 7 Data Vector*			

Sidband Data:

```
Set # Amplifier Protection Switch on messageBlanking
Set 1: Stimulatelel Switch
Set 2: SGS Bit Selection
Set 3: SGS Bit-15
```

Data Vector decoding:

```
Bit 31: Reserved
Bit 30: DAC/SBS Data Vector*
Bit 29: DAC/SBS Data Vector*
Bit 28: Long Loop Prt. Vector
Bit 27: Short Loop Prt. Vector
Bit 26: Reserved
Bit 25: Reserved
Bit 24: Reserved
Bit 23: Reserved
Bit 22: Reserved
Bit 21: Reserved
Bit 20: Reserved
Bit 19: Reserved
Bit 18: Reserved
Bit 17: Reserved
Bit 16: Reserved
Bit 15: Reserved
Bit 14: Reserved
Bit 13: Reserved
Bit 12: Reserved
Bit 11: Reserved
Bit 10: Reserved
Bit 9: Reserved
Bit 8: Reserved
Bit 7: Reserved
Bit 6: Reserved
Bit 5: Reserved
Bit 4: Reserved
Bit 3: Reserved
Bit 2: Reserved
Bit 1: Reserved
Bit 0: Reserved

DAC/SBS Data Vector(0x00):
Bit 31: Reserved
Bit 30: Repeat Timeline (0: 20 ms, 1: 1000*20ms)
Bit 29: DAC/SBS Data Vector*
Bit 28: DAC/SBS Data Vector*
Bit 27: Long Loop Prt. Vector
Bit 26: Short Loop Prt. Vector
Bit 25: Reserved
Bit 24: Reserved
Bit 23: Reserved
Bit 22: Reserved
Bit 21: Reserved
Bit 20: Reserved
Bit 19: Reserved
Bit 18: Reserved
Bit 17: Reserved
Bit 16: Reserved
Bit 15: SGS Bit-15 (DAC value (whichever is bit value, 0xFFFF is zero level) / SGS data value)
Bit 14: SGS Bit Selection (Amplifier Protection Switch/Blanking)
Bit 13: SGS Bit-15 (Stimulation)
Bit 12: SGS Bit-15 (Electrode Config ID)
Bit 11: SGS Bit-15 (Electrode Config ID)

Loop Prt. Vector(0x00):
Bit 31: Reserved
Bit 30: Reserved
Bit 29: Number of Repeat (2: Vectors are repeated once, thus used twice)
Bit 28: Number of Repeat (2: Vectors are repeated once, thus used twice)
Bit 27: Loop Prt. Vector (Number of vectors to jump backand)
Bit 26: Loop Prt. Vector (Number of vectors to jump backand)
Bit 25: Loop Prt. Vector (Number of vectors to jump backand)
Bit 24: Loop Prt. Vector (Number of vectors to jump backand)
Bit 23: Loop Prt. Vector (Number of vectors to jump backand)
Bit 22: Loop Prt. Vector (Number of vectors to jump backand)
Bit 21: Loop Prt. Vector (Number of vectors to jump backand)
Bit 20: Loop Prt. Vector (Number of vectors to jump backand)
Bit 19: Loop Prt. Vector (Number of vectors to jump backand)
Bit 18: Loop Prt. Vector (Number of vectors to jump backand)
Bit 17: Loop Prt. Vector (Number of vectors to jump backand)
Bit 16: Loop Prt. Vector (Number of vectors to jump backand)
Bit 15: Loop Prt. Vector (Number of vectors to jump backand)
Bit 14: Loop Prt. Vector (Number of vectors to jump backand)
Bit 13: Loop Prt. Vector (Number of vectors to jump backand)
Bit 12: Loop Prt. Vector (Number of vectors to jump backand)
Bit 11: Loop Prt. Vector (Number of vectors to jump backand)
Bit 10: Loop Prt. Vector (Number of vectors to jump backand)
Bit 9: Loop Prt. Vector (Number of vectors to jump backand)
Bit 8: Loop Prt. Vector (Number of vectors to jump backand)
Bit 7: Loop Prt. Vector (Number of vectors to jump backand)
Bit 6: Loop Prt. Vector (Number of vectors to jump backand)
Bit 5: Loop Prt. Vector (Number of vectors to jump backand)
Bit 4: Loop Prt. Vector (Number of vectors to jump backand)
Bit 3: Loop Prt. Vector (Number of vectors to jump backand)
Bit 2: Loop Prt. Vector (Number of vectors to jump backand)
Bit 1: Loop Prt. Vector (Number of vectors to jump backand)
Bit 0: Loop Prt. Vector (Number of vectors to jump backand)

Long Loop Prt. Vector(0x00):
Bit 27 = 0 : Address Offset (Number of vectors to jump backand)
Bit 26 = 0 : Number of Repeats

Loop One Vector(0x00):
Bit 27 - 0 : Number of Repeats

End Command[11]:
Bit 31: Reserved
Bit 30: Reserved
Bit 29: Reserved
Bit 28: Reserved
Bit 27: Reserved
Bit 26: Reserved
Bit 25: Reserved
Bit 24: Reserved
Bit 23: Reserved
Bit 22: Reserved
Bit 21: Reserved
Bit 20: Reserved
Bit 19: Reserved
Bit 18: Reserved
Bit 17: Reserved
Bit 16: Reserved
Bit 15: Reserved
Bit 14: Reserved
Bit 13: Reserved
Bit 12: Reserved
Bit 11: Reserved
Bit 10: Reserved
Bit 9: Reserved
Bit 8: Reserved
Bit 7: Reserved
Bit 6: Reserved
Bit 5: Reserved
Bit 4: Reserved
Bit 3: Reserved
Bit 2: Reserved
Bit 1: Reserved
Bit 0: Reserved
```

HS <-> STG interconnection (bits 32-0)

STG HS connector					
Pin	Name	Bit	HS FPGA Pin	Function	to FPGA: STG FPGA Pin
3	STG01	0	V1	Reset	X2
4	STG02	1	V2	Suspend	X2
5	STG03	2	T3	Reserved	X2
6	STG04	3	U1	Reserved	X2
7	STG05	4	T1	Reserved	X2
8	STG06	5	V3	Reserved	X2
9	STG07	6	P3	Reserved	X2
10	STG08	7	R1	Blanking	X2
11	STG09	8	V3	50 kHz Impulse	X2
12	STG10	9	W1	SPI: read data ready	X2
13	STG11	10	Y1	SPI: SCLK	X2
14	STG12	11	V2	SPI: CS	X2
15	STG13	12	W3	SPI: MOSI	X2
16	STG14	13	A4	SPI: MISO	X2
17	STG15	14	A2A	25 MHz clock	X1,X2
18	STG16	15	Y3	400 kHz Power sync.	X1,X2
19	STG17	16	H4	400 kHz Power sync.	X1,X2
20	STG18	17	H4	38.4 MHz clock	X1,X2
21	STG19	18	G3	Reset	X1
22	STG20	19	H8	Suspend	X1
23	STG21	20	G6	Reserved	X1
24	STG22	21	G4	Reserved	X1
25	STG23	22	F5	Reserved	X1
26	STG24	23	G7	Reserved	X1
27	STG25	24	H5	Reserved	X1
28	STG26	25	J7	Blanking	X1
29	STG27	26	J3	50 kHz	X1
30	STG28	27	J4	SPI: read data ready	X1
31	STG29	28	J6	SPI: SCLK	X1
32	STG30	29	K4	SPI: CS	X1
33	STG31	30	K5	SPI: MOSI	X1
34	STG32	31	K6	SPI: MISO	X1

Abgerufen von „http://wiki.mcs.de.com/index.php?title=MEA2100_User_Guide&oldid=19747“

■ Diese Seite wurde zuletzt am 19. November 2015 um 13:41 Uhr geändert