

# DAQ data structure for the Muon g-2 experiment

Wes Gohn, Tim Gorringe, Ran Hong, Kim Siang Khaw\*, David Sweigart

March 12, 2017

## Abstract

This document outlines the DAQ data structure of the Muon g-2 experiment. A detailed list of the MIDAS data bank will be shown and their contents will be described.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Data unpacking in a nutshell . . . . .	2
1.2	MIDAS DAQ output in a nutshell . . . . .	2
<b>2</b>	<b>MIDAS Bank list</b>	<b>3</b>
2.1	Calorimeter-related banks . . . . .	3
2.2	Auxiliary detector-related banks . . . . .	3
2.3	CCC related banks . . . . .	3
2.4	Field related banks . . . . .	4
<b>3</b>	<b>Bank contents</b>	<b>5</b>
3.1	Calorimeter-related banks . . . . .	5
3.2	Auxiliary detector-related banks . . . . .	9
3.3	CCC related banks . . . . .	10
3.4	Field related banks . . . . .	10
<b>4</b>	<b>Parsers for MIDAS bank data</b>	<b>16</b>
<b>5</b>	<b>Cheat sheet for data unpacking</b>	<b>16</b>

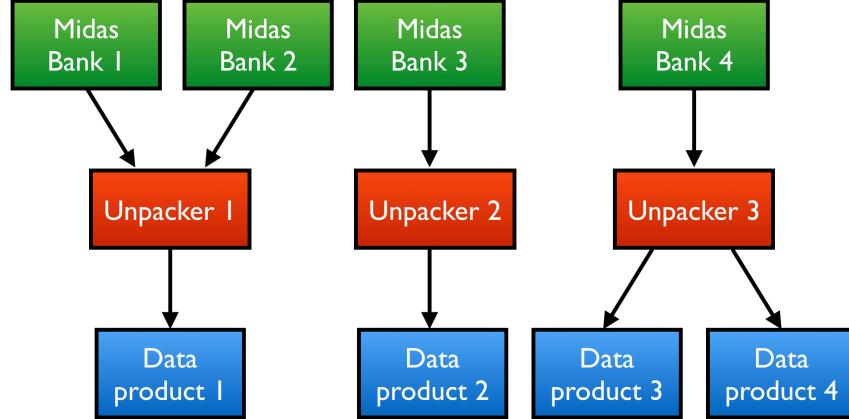
---

\*Corresponding author, khaw84@uw.edu

# 1 Introduction

## 1.1 Data unpacking in a nutshell

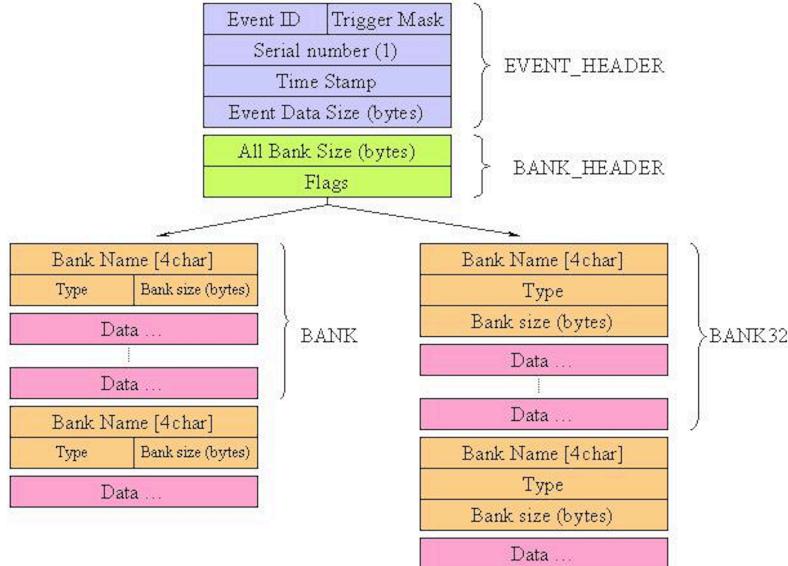
The basic idea of data unpacking is to convert compact DAQ information stored in MIDAS banks to well-structured data products, as depicted in Fig. 1. Several unpackers are used to decode the information stored in MIDAS banks and the outputs are stored in different data products which are basically structs in C++ language. A complete list of the Midas bank - Data unpacker - Data product relationship is summarized in Sec. 5.



**Figure 1:** Simplified diagram of the flow of the data unpacking.

## 1.2 MIDAS DAQ output in a nutshell

The main DAQ framework for the Muon g-2 experiment is based on MIDAS [1]. MIDAS event structure is as depicted in Fig. 2.



**Figure 2:** MIDAS event structure. Each event has its header that is followed by the bank header. Then all the banks will appear according the defined order.

## 2 MIDAS Bank list

Hundred of banks will be stored in each MIDAS event and it is very important to classify them properly. At the moment they can be grouped into 4 categories: calorimeter, auxiliary detector, CCC and magnetic field. Naming of these banks will be described in this section and their contents will be explained in the next section.

### 2.1 Calorimeter-related banks

There are 3 fill types for the calorimeter. Muon fill is the typical muon events, laser fill is event dedicated for laser calibration and monitoring events and pedestal fill is trivia from its name. Data from each fill type is identified from the bank name. The muon fill is denoted by “C”, the laser fill is denoted by “L” and the pedestal fill is denoted by “P”. A summary of the banks is listed in Tab. 1.

**Table 1:** *MIDAS bank list for the calorimetry data.*

muon fill	laser fill	pedestal fill	Description
Bank name			
CA	LA	PA	AMC13 Header
CB	-	-	WFD5 header
CC	LC	PC	GPU timing data
CF	-	-	GPU fitted data
CH	-	-	per-crystal Q-method data (N-th event, end of run)
CL	LL	PL	Clock data
CP	-	-	Pedestal
CQ	-	-	per-calorimeter Q-method data (every event)
CR	LR	PR	WFD5 raw data
CT	-	-	T-method islands
CZ	LZ	PZ	AMC13 CDF trailers

Since each bank has to be named exactly four characters, the last 2 characters denoted the calorimeter number, e.g. **CT03** means the T-method bank from calorimeter number 3. Hence the number 01-24 are reserved for the calorimeters and 25 is reserved for the laser system.

### 2.2 Auxiliary detector-related banks

Systems that fall into the auxiliary detector category are the kickers, quadrupoles, IBMS, T0 counter and fiber harps. Depending on the length of the signal, two different types of digitizers will be used to digitize these signals: Cornell WFD5 and CANE 1742 digitizers. A separate T/Q-method is needed for auxiliary detectors. Their data banks are denoted with the initial “K”. A list of these banks are summarized in Tab. 2. The last 2 characters of the bank name will be decided soon.

### 2.3 CCC related banks

This is the bank storing the information regarding the CCC system based on FC7. A list of these banks are summarized in Tab. 3.

**Table 2:** MIDAS bank list for auxiliary T/Q data. This is mainly for the fiber harps, quads, kickers and IBMS.

Bank name	Description
KH	Per aux. detector channel Q-method data (N-th event, end of run)
KQ	Per aux. detector Q-method data (every event)
KT	T-method data
IBMS	IBMS waveforms

**Table 3:** MIDAS bank list for the CCC data.

TTCA	AMC13 Header
TTCR	CCC AMC13 Payload
TTCZ	AMC13 Trailer

## 2.4 Field related banks

All field-team banks are filled once per event and the definition of an event is different from the one of the  $\omega_a$  related banks. For many field-team banks, a c struct is defined in the `field_struct.hh` file, accessible for all frontends and unpackers. Programmers should able to cast the read-out bank (array of bytes) onto a pointer of the corresponding struct. A midas bank can be an entire struct (like **TLNP**, **ABPR**, etc) or a array of structs (like **GALI**). A list of these banks are summarized in Tab. 4.

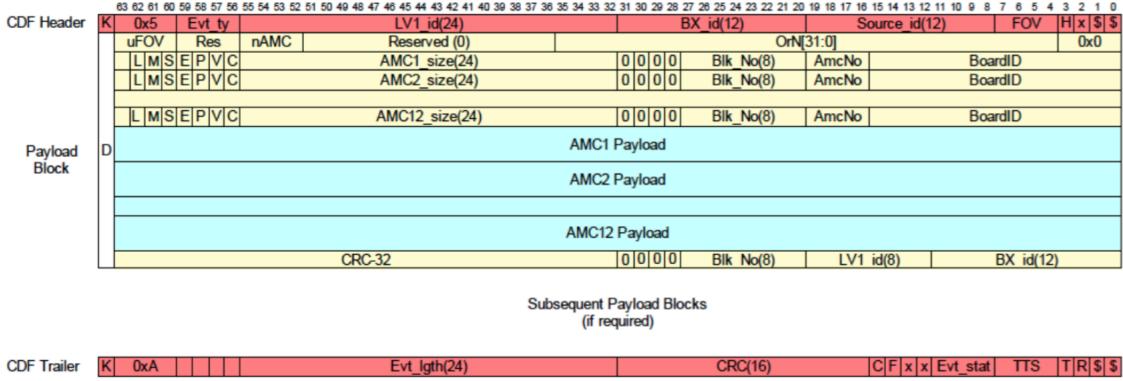
**Table 4:** MIDAS bank list for the magnetic field related data.

System	Name	Description
Fixed probe	FXPR	Fixed probe, header + NMR waveforms
Trolley	TLNP	Trolley NMR Pulse, header + NMR waveforms
	TLBC	Trolley Barcode, header + Barcode waveforms
	TLMN	Trolley Monitors (temperatures, voltages and pressure), header + voltage waveforms
	GALI	Galil (trolley and garage) data, positions + velocities + control voltages + tensions
Absolute probes	ABPR	Absolute probe (spherical probe and plunging probe are using the same bank), header + NMR waveforms
Flux gate	FLUX	Flux gate, fluxgate waveforms
Surface coil	SFCL	Surface coil, current readouts

### 3 Bank contents

This section details contents of each MIDAS bank. All the banks are packed in 32-bit word integer regardless of the original format.

#### 3.1 Calorimeter-related banks



**Figure 3:** Data structure for AMC13 to DAQ. The first two 64-bit words are stored in the CA (LA, PA) banks and the last two 64-bit words are stored in the CZ (LZ, PZ) banks.

#### CA (LA, PA) banks

This is the bank for the AMC13 to DAQ header information. The first 64-bit word is the CDF header and the next 64-bit word is the payload header as shown in Fig. 3.

#### CZ (LZ, PZ) banks

This is the bank for the AMC13 to DAQ trailer information. The first 64-bit word is the last 64-bit word of the payload block and the next 64-bit word is the CDF trailer as shown in Fig. 3.

#### CB banks

This is the bank for the WFD5 to AMC13 header information as shown in Fig. 4.



**Figure 4:** Data structure for Rider to AMC13.

#### CR (LR, PR) banks

This is the bank for the full WFD5 payload as shown in Fig. 5. Due to the huge data size of raw payload, a pre-scale of about 1000 event is usually being set in the MIDAS ODB.

	63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 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
Channel Header	0 1 Channel Tag [15:0] Waveform Gap [21:0] Waveform Count [11:0] DDR3 Start Address [25:14]
Channel Header	DDR3 Start Address [13:0] Waveform Length [22:0] FT Trigger Number [23:0]
Waveform 1 Header	Waveform Count [11:0] DDR3 Start Address [25:0] FT Waveform Length [22:0]
Waveform 1 Header	0 1 0 Channel Tag [15:0] Waveform Gap [21:0] Waveform Index [11:0]
	Waveform 1 ADC Data
Waveform 2 Header	Waveform Count [11:0] DDR3 Start Address [25:0] FT Waveform Length [22:0]
Waveform 2 Header	0 1 0 Channel Tag [15:0] Waveform Gap [21:0] Waveform Index [11:0]
	Waveform 2 ADC Data
	...
Waveform N Header	Waveform Count [11:0] DDR3 Start Address [25:0] FT Waveform Length [22:0]
Waveform N Header	0 1 0 Channel Tag [15:0] Waveform Gap [21:0] Waveform Index [11:0]
	Waveform N ADC Data
Channel Trailer	Channel Checksum
Channel Trailer	Channel Checksum
Channel Trailer	Data Integrity Check Data Transfer Time

**Figure 5:** Data structure for the WFD5 raw payload.

## CT banks

This is the bank for calorimeter T-method chopped islands. A collection of 54 islands from 54 crystals/segments are extracted per trigger. The detailed structure is shown in Fig. 6.

Bank : CT04 Length: 243196(I*1)/60799(I*4)/121598(Type) Type:Signed Integer*2									
1 ->	-9480	1	23	54	27	0	13554	0	
9 ->	98	0	1119	1129	1129	1125	1120	1148	
17 ->	1134	1197	1182	1531	2046	2046	2046	2046	
25 ->	1930	1507	1352	1285	1237	1223	1210	1197	
33 ->	1190	1175	1160	1163	1164	1158	1143	1151	
41 ->	1159	1159	1149	1140	1137	1150	1143	1143	
49 ->	1143	1140	1142	1149	1128	1135	1138	1134	
57 ->	1138	1135	1125	1135	1133	1131	1129	1128	
65 ->	1137	1140	1136	1134	1132	1141	1130	1128	

array of signed tow-byte integers

32-bit number of 16-bit words in bank (the above entry maps to 0x0001dafe = 121598)

16-bit number of islands

16-bit number of segments

32-bit CTAG /TBD

number of islands x (

32-bit island time +

32-bit island length +

+ number of segments \* length of island \* 16-bit ADC samples )

**Figure 6:** Data structure for the CT bank (T-method chopped islands).

## CH banks

This is the bank for calorimeter segment histograms. Histogram from each segment is being summed to itself each fill and is flushed out every N-th event. The detailed structure is shown in Fig. 7.

## CQ banks

This is the bank for calorimeter sum histograms. Histograms from all segments are summed together and decimated, and then are being flushed out every fill. The detailed structure is

Bank:KH01 Length: 508(I*1)/127(I*4)/127(Type) Type:Unsigned Integer*4
1-> 0x00000078 0x00000001 0x00000200 0x00000004 0x00000002 0x00000001 0x000000e10
9-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
17-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
25-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
33-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
41-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
49-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
57-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10
65-> 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000000e10 0x000001c20
73-> 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20
81-> 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20
89-> 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20
97-> 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20 0x000001c20
105-> 0x00003840 0x00003840 0x00003840 0x00003840 0x00003840 0x00003840 0x00003840
113-> 0x00003840 0x00003840 0x00003840 0x00003840 0x00003840 0x00003840 0x00003840
121-> 0x00007080 0x00007080 0x00007080 0x00007080 0x00007080 0x00007080 0x00007080

CH databank words are signed 32-bit signed integers

- first word - number of array elements of Q method histogram
- second word - first ADC sample within fill of Q-method histogram (is an ODB parameter)
- third word - last ADC sample within fill of Q-method histogram (is an ODB parameter)
- fourth word – initial rebin factor of first sub-histogram / rebin interval (is an ODB parameter)
- fifth word – number of sub-histograms with different rebin factors (is an ODB parameter)
- sixth word – rebin multiplier for  $i$ th sub hist,  $\text{rebin}_i = \text{rebin}_0 * \text{multiplier}^i$  (ODB parameter)
- seventh word - number of segments / detectors in histogram (derived from ODB parameters)
- remaining words - Q-method histogram array elements of size specified by first word

**Figure 7:** Data structure for the CH bank (calo segment histograms). In this example, the number of samples = 512, initial rebin = 2, rebin intervals = 4 and rebin multiplier = 2.

shown in Fig. 8.

Bank:CQ04 Length: 70004(I*1)/17501(I*4)/17501(Type) Type:Unsigned Integer*4
1-> 0x0000445d 0xfffffffff9 0xfffffff8a 0x00000025 0xfffffff78 0xfffffffbb 0x0000002a 0x0000000b
9-> 0x0000004d 0xfffffff9d 0x0000008a 0x0000007b 0x000000b7 0x000000a 0x00000048 0x000000ee
17-> 0xfffffff8 0x0000002c 0x00000022 0x00000024 0x000000a1 0x0000005a 0x00000041 0x0000007e
25-> 0x00000042 0x00000028 0x000000f6 0x0000003f 0x000000fe 0x0000007f 0x000000c0 0x00000056
33-> 0x0000009a 0x00000082 0x00000067 0x0000012c 0x000000cc 0x00000064 0x00000077 0x00000044
41-> 0xfffffff9b 0xfffffff1 0x00000011 0x000000a7 0x0000004a 0x0000001c 0x00000065 0x00000021

(number of histogram array elements + 1) x signed four-byte integers

total number of data words, i.e. histogram array elements + 1  
segment summed, time-decimated, pedestal subtracted histogram array elements

**Figure 8:** Data structure for the CQ bank (calo sum histograms).

## CP banks

This is the bank for calorimeter pedestals. The detailed structure is shown in Fig. 9.

## CC (LC, PC) banks

This is the bank for the calorimeter DAQ performance related information. It has information like tcp timing and gpu timing. The detailed structure is shown in Fig. 10.

```

Bank:CP04 Length: 220(I*1)/55(I*4)/55(Type) Type:Real*4 (FMT machine dependent)
1-> 5.400e+01 1.126e+03 1.293e+03 1.301e+03 1.328e+03 1.329e+03 1.780e+03 1.761e+03
9-> 1.761e+03 1.768e+03 1.781e+03 1.774e+03 1.761e+03 1.751e+03 1.780e+03 1.781e+03
17-> 1.764e+03 1.736e+03 1.725e+03 1.711e+03 1.767e+03 1.779e+03 1.751e+03 1.759e+03
25-> 1.768e+03 1.760e+03 1.767e+03 1.752e+03 1.764e+03 1.772e+03 1.765e+03 1.753e+03
33-> 1.754e+03 1.752e+03 1.783e+03 1.780e+03 1.760e+03 1.747e+03 1.736e+03 1.779e+03
41-> 1.767e+03 1.753e+03 1.758e+03 1.730e+03 1.755e+03 1.771e+03 1.799e+03 1.765e+03
49-> 1.779e+03 1.752e+03 1.794e+03 1.753e+03 1.753e+03 1.759e+03 1.742e+03

```

(number of segments + 1) x four bytes float format

number of segments  
number of segments x pedestal values

**Figure 9:** Data structure for the CP bank (T-method pedestals).

```

Bank:CC04 Length: 152(I*1)/38(I*4)/38(Type) Type:Unsigned Integer*4
1-> 0x2cf01551 0x0800c0f3 0x584127e1 0x00000000 0x000913c3 0x00000000 0x584127e1 0x00000000
9:- 0x000913c4 0x00000000 0x584127e1 0x00000000 0x0009e7b6 0x00000000 0x584127e1 0x00000000
17:- 0x000a1d59 0x00000000 0x584127e1 0x00000000 0x000a0be5 0x00000000 0x584127e1 0x00000000
25-> 0x000a1d58 0x00000000 0x584127e1 0x00000000 0x000a1d76 0x00000000 0x584127e1 0x00000000
33-> 0x000a1dce 0x00000000 0x00000e75 0x00000000 0x00000e75 0x00000000

```

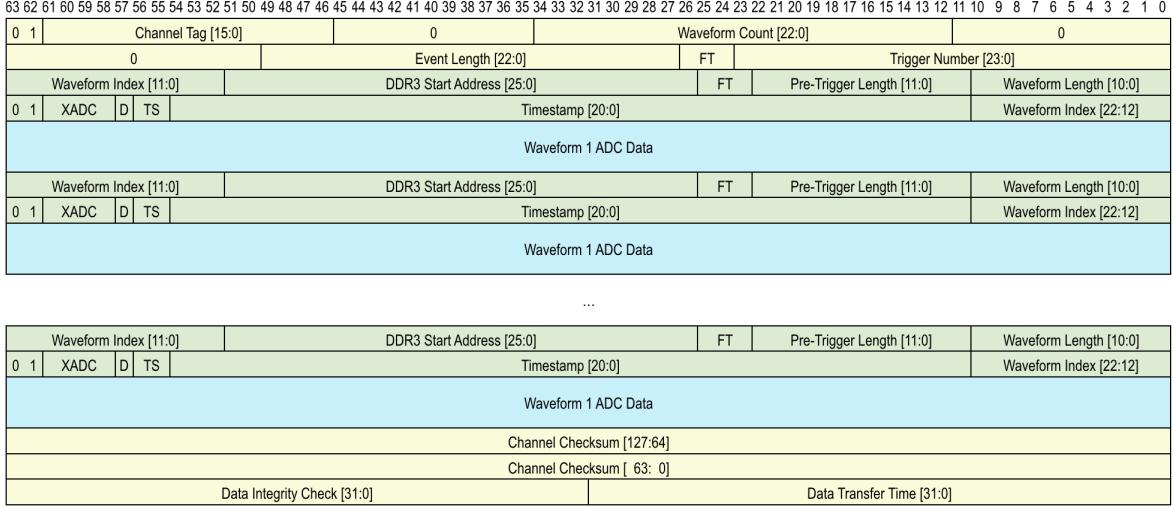
array of 64-bit words (sec, usecs are obtained from gettimeofday() and struct timeval in sys/time.h)

64-bit CDF header word  
TCP proc unlocked / started, first 64-bit word is seconds, second 64-bit word is usecs  
got TCP header word, first 64-bit word is seconds, second 64-bit word is usecs  
got TCP header word, first 64-bit word is seconds, second 64-bit word is usecs  
GPU proc unlocked / started , first 64-bit word is seconds, second 64-bit word is usecs  
GPU copy done , first 64-bit word is seconds, second 64-bit word is usecs  
GPU proc done , first 64-bit word is seconds, second 64-bit word is usecs  
MFE proc unlocked, first 64-bit word is seconds, second 64-bit word is usecs  
MFE banks made, first 64-bit word is seconds, second 64-bit word is usecs  
current TCP fill number  
current GPU fill number

**Figure 10:** Data structure for the CC bank (calo performance).

## CR (LR, PR) banks, asynchronous

This is the bank for the WFD5 payload in the asynchronous mode. The detailed structure is shown in Fig. 11.



**Figure 11:** Data structure for WFD5 asynchronous mode.

## CF banks

This is the bank for GPU fitted results. The bank starts with an “unsigned int” that is the number of segments. Then for each of those segments there is a block like this: an “unsigned int” which is the number of pulses and then *nPulses* blocks of *pulseFinderResult* struct. The member variables in this struct is shown in Tab. 5.

**Table 5:** Format of the struct *pulseFinderResult*.

start word index	type	field name	content
0	Float_t	time	fitted time
2	Float_t	phase	fitted phase
4	Float_t	energy	fitted energy
6	Float_t	pedestal	fitted pedestal
8	Float_t	chi2	chi square of the fit
10	Float_t	pedestal	fitted pedestal
12	Int_t	peak_index	index of the fitted peak
14	Int_t	peak_value	value of the fitted peak

## 3.2 Auxiliary detector-related banks

This section will be updated once decision is finalized.

### KH and KQ banks

These two banks have the same format as the CH and CQ banks.

## KT bank

This bank has the same format as the CT bank.

## IBMS

This bank will store all the waveforms from CAEN1742. Actual format to be decided.

### 3.3 CCC related banks

#### TTCA, TTCR, TTCZ banks

TTCA bank will have the same format as the CA bank in the calorimeter-related data since the uTCA crate has the same data format. TTCZ on the other hand will have the same format as the CZ bank.

There are two types FC7 – encoder and fanout, and they share the same data format. The data format of the encoder and fanout FC7 are shown in Fig. 12.

AMC13 Header	0   0   0   0   AMC #	Trigger # [23:0]	Timestamp [43:32]	Data Length [19:0]
AMC13 Header	E   0   0   0   0   0   XADC	TT [4:0]	Timestamp [31:0]	BT   F   Board ID [11:0]
FC7 Data	Pre-Trigger Gap [31:0]			0   0   0   0   0   0   0   S[3:0]   TI[3:0]   L8 FMC ID [7:0]   L12 FMC ID [7:0]
FC7 Data	Output Trigger A Types [31:0]			Output Trigger B Width   Overflow Warning Threshold [23:0]
FC7 Data	Output Trigger B Delay [31:0]			Output Trigger A Delay [31:0]
FC7 Data	0   0   0   0   0   0   S   0   X   T   R   0   0   A   A   L8 RX LOS Error   L12 RX LOS Error	L8 TX Fault Error	L12 TX Fault Error	L8 Mod. Abs. Error   L12 Mod. Abs. Error
FC7 Data	TTS Lock Threshold [31:0]			L8 TTS Locks   L12 TTS Locks   L8 Enabled Ports   L12 Enabled Ports
FC7 Data	TTC Delay 8 [7:0]	TTC Delay 7 [7:0]	TTC Delay 6 [7:0]	TTC Delay 5 [7:0]
FC7 Data	TTC Delay 16 [7:0]	TTC Delay 15 [7:0]	TTC Delay 14 [7:0]	TTC Delay 13 [7:0]
FC7 Data	Transceiver Serial Number 1 [63:0]			TTC Delay 4 [7:0]   TTC Delay 3 [7:0]   TTC Delay 2 [7:0]   TTC Delay 1 [7:0]
FC7 Data	Transceiver Serial Number 1 [127:64]			TTC Delay 12 [7:0]   TTC Delay 11 [7:0]   TTC Delay 10 [7:0]   TTC Delay 9 [7:0]
FC7 Data	Transceiver Serial Number 8 [63:0]			
FC7 Data	Transceiver Serial Number 8 [127:64]			
FC7 Data	Transceiver Serial Number 9 [63:0]			
FC7 Data	Transceiver Serial Number 9 [127:64]			
FC7 Data	Transceiver Serial Number 15 [63:0]			
FC7 Data	Transceiver Serial Number 15 [127:64]			
AMC13 Trailer	CRC [31:0]		Trigger # [7:0]   0   0   0   0	Data Length [19:0]

Figure 12: Data structure for both encoder and fanout FC7.

### 3.4 Field related banks

#### FXPR bank

This is the bank for fixed probes. It consists of a header and NMR waveforms. Structure of the FXPR bank is summarized in Tab. 6 and its macros are summarized in Tab. 7.

#### TLNP bank

This is the bank for Trolley NMR pulses. It consists of a header and NMR waveforms. Structure of the TLNP bank is summarized in Tab. 8 and its macros are summarized in Tab. 9.

**Table 6:** *MIDAS bank structure for the FXPR bank.*

start word index	type	array length	field name	content	struct name
0	Double_t	num_ch	sys_clock	system clock	fixed_t
4*num_ch	Double_t	num_ch	gps_clock	gps clock	
8*num_ch	Double_t	num_ch	dev_clock	device clock	
12*num_ch	Double_t	num_ch	snr	signal to noise ratio	
16*num_ch	Double_t	num_ch	len	length of each wave form	
20*num_ch	Double_t	num_ch	freq	frequency extracted	
24*num_ch	Double_t	num_ch	ferr	frequency error	
28*num_ch	Double_t	num_ch	freq_zc	frequency extracted, zero crossing	
32*num_ch	Double_t	num_ch	ferr_zc	frequency error, zero crossing	
36*num_ch	UShort_t	num_ch	health	health indicator of probes	
37*num_ch	UShort_t	num_ch	method	frequency extraction method	
38*num_ch	UShort_t	num_ch * rec_len	trace	NMR waveforms: Waveform_Ch1 + Waveform_Ch2 + ... + Waveform_Ch6	

**Table 7:** *Hard-coded macros in the FXPR bank.*

Name in the code	Name in this doc	Value
NMR_NUM_FIXED_PROBES	num_ch	378
NMR_FID_LENGTH_RECORD	rec_len	10000

**Table 8:** *MIDAS bank structure for the TLNP bank.*

start word index	type	array length	field name	content	struct name
0	ULong64_t	1	gps_clock	Time stamp of the first NMR sample	trolley_nmr_t
4	UShort_t	1	probe_index	probe index	
5	UShort_t	1	length	length of the NMR waveform	
6	Short_t	nmr_len	trace	Trolley Probe NMR wavefrom	

**Table 9:** *Hard-coded macros in the TLNP bank.*

Name in the code	Name in this doc	Value
TRLY_NMR_LENGTH	nmr_len	24000

## TLBC bank

This is the bank for Trolley barcode readers. It consists of a header and barcode waveforms. Structure of the TLBC bank is summarized in Tab. 10 and its macros are summarized in Tab. 11.

**Table 10:** *MIDAS bank structure for the TLBC bank.*

start word	index	type	array length	field name content	struct name
0	ULong64_t	1	gps_clock	Time stamp of the first barcode sample	trolley_barcode_t
4	UShort_t	1	length_per_ch	length of the barcode waveform per channel	
5	UShort_t	bc_ch*bc_len	traces	Barcode wavefroms: Waveform_Ch1 + Waveform_Ch2 + ... + Waveform_Ch6	

**Table 11:** *Hard-coded macros in the TLBC bank.*

Name in the code	Name in this doc	Value
TRLY_BARCODE_LENGTH	bc_len	3000
TRLY_BARCODE_CHANNELS	bc_ch	6

## TLMN bank

This is the bank for Trolley monitors. It consists of information like temperatures, voltages and pressures. Structure of the TLMN bank is summarized in Tab. 12 and its macros are summarized in Tab. 13.

## GALI bank

This is the bank for trolley and garage data. It has information like the trolley positions and velocities. Structure of the GALI bank is summarized in Tab. 14 and 15, and its macros are summarized in Tab. 16.

## ABPR bank

This is the bank for absolute probe data. Both spherical probe and plunging probe are using the same bank. It has information like the header and NMR waveforms. Structure of the ABPR bank is summarized in Tab. 17.

## FLUX bank

This is the bank for the fluxgate. It has information like the fluxgate waveforms. Structure of the FLUX bank is summarized in Tab. 18 and its macros are summarized in Tab. 19.

---

<sup>1</sup>Note: the length of the NMR waveform is not fixed from event to event. For each event, its length is determined by the "length" field of the "absolute\_nmr\_info\_t" struct in the header section.

**Table 12:** MIDAS bank structure for the TLMN bank.

start word index	type	array length	field name	content	struct name
0	ULong64_t	1	gps_clock_cycle_start	Time stamp of the measurement cycle	trolley_monitor_t
4	UInt_t	1	PMonitorVal	Pressure Monitor readout value	
6	Uint_t	1	PMonitorTemp	Pressure Monitor temperature	
8	Uint_t	1	RFPower1	RF Power monitor 1	
10	Uint_t	1	RFPower2	RF Power monitor 2	
12	Uint_t	1	NMRCheckSum	NMR waveform check sum, calculated by trolley interface	
14	Uint_t	1	FrameCheckSum	Frame data check sum, calculated by trolley interface	
16	Uint_t	1	FrameSum	Frame data sum calculated by DAQ	
18	Uint_t	1	FrameIndex	Frame index	
20	UShort_t	1	StatusBits	(Bit 5-7)Reserved	
21	UShort_t	1	TMonitorIn	Temperature monitor, interia	
22	UShort_t	1	TMonitorExt1	Temperature monitor, exteria 1	
23	UShort_t	1	TMonitorExt2	Temperature monitor, exteria 2	
24	UShort_t	1	TMonitorExt3	Temperature monitor, exteria 3	
25	UShort_t	1	V1Min	Voltage Monitor 1, minimum	
26	UShort_t	1	V1Max	Voltage Monitor 1, maximum	
27	UShort_t	1	V2Min	Voltage Monitor 2, minimum	
28	UShort_t	1	V2Max	Voltage Monitor 2, maximum	
29	UShort_t	1	length_per_ch	waveform length per channel for the voltage monitors	
30	UShort_t	mn_len	trace_VMonitor1	waveform of voltage monitor 1	
30+mn_len	UShort_t	mn_len	trace_VMonitor2	waveform of voltage monitor 2	

**Table 13:** Hard-coded macros in the TLMN bank.

Name in the code	Name in this doc	Value
TRLY_MONITOR_LENGTH	mn_len	3000

**Table 14:** MIDAS bank structure for the GALI bank.

start word index	type	array length	field name	content	struct name
0	galil_data_t	group_size	-	Array of structs: galil_data_t	galil_data_t

**Table 15:** Format of the struct galil\_data\_t.

start word index	type	array length	field name	content
0	ULong64_t	1	TimeStamp	time stamp of the readout
4	Int_t	2	Tensions	Tensions of the fishing line and the signal cable
8	Int_t	3	Positions	Positions: Motor 1, Motor 2 and Garage
14	Int_t	3	Velocities	Velocities: Motor 1, Motor 2 and Garage
20	Int_t	3	OutputVs	Output voltages: Motor 1, Motor 2 and Garage

**Table 16:** Hard-coded macros in the GALI bank.

Name in the code	Name in this doc	Value
GALILREADGROUPSIZE	group_size	50

**Table 17:** MIDAS bank structure for the ABPR bank.

start word index	type	array length	field name	content	struct name
0	ULong64_t	1	time_stamp	Time stamp of the NMR sample	absolute_nmr_info_t
4	Uint_t	1	length	probe index	
6	Uint_t	4	position	length of the NMR waveform	
14	UShort_t	1	flay_run_number	Trolley Probe NMR waveform	
15	UShort_t	1	probe_index		
16	UShort_t	length	-	NMR waveform <sup>1</sup>	

**Table 18:** *MIDAS bank structure for the FLUX bank.*

start word index	type	array length	field name	content	struct name
0	float64	num_ch*period*rate	-	Flux gate wave forms for all channels: Waveform_ch1 + Waveform_ch2 + ...	

**Table 19:** *Hard-coded macros in the FLUX bank.*

Name in the code	Name in this doc	Value
FLUX_NUM_CHANNELS	num_ch	9
FLUX_TRACE_PERIOD	period	60
FLUX_RATE	rate	8000

## SFCL bank

This is the bank for the surface coil. It has the information like current readouts from the surface coils. Structure of the SFCL bank is summarized in Tab. 20 and its macros are summarized in Tab. 21.

**Table 20:** *MIDAS bank structure for the SFCL bank.*

start word index	type	array length	field name	content	struct name
0	Double_t	num_coil	sys_clock	system clock	surface_coil_t
4*num_coil	Double_t	num_coil	gps_clock	gps clock	
8*num_coil	Double_t	num_coil	top_board	top board currents	
12*num_coil	Double_t	num_coil	bot_board	bottom board currents	

**Table 21:** Hard-coded macros in the SFCL bank.

Name in the code	Name in this doc	Value
SC_NUM_COILS	num_coil	100

## 4 Parsers for MIDAS bank data

Muon g-2 offline analysis framework relies on parsers in the gm2parser namespace hosted under repository gm2unpackers to decode the data. To checkout the codes,

```
git clone ssh://p-gm2unpackers@cdcv.s.fnal.gov/cvs/projects/gm2unpackers
```

Alternatively, you can also use

```
mrb g gm2unpackers
```

in our g-2 environment. These parsers are written in C++ and are being used in the *art* producer modules. They can also be used in your standalone C++ codes, if you wish to. The parsers are located at

```
gm2unpackers/calo/parsers
```

at the moment for both calorimeter and CCC related information. This section will be updated accordingly once they are reorganized.

## 5 Cheat sheet for data unpacking

A complete list of midas bank to unpacker to data product is summarized in Tab. 22.

**Table 22:** Midas bank-Data unpacker-Data product relationship table for a muon fill. This is the mapping as of Jan 10, 2017 and will be updated accordingly once the restructuring of the data unpacking is completed.

Midas bank	Data unpacker ( <i>art</i> producer module)	Data product ( <i>struct</i> )
CR	RawBankUnpacker	gm2calo::RiderArtRecord gm2calo::AMCHeader gm2calo::RiderChannelHeader gm2calo::RiderWaveformHeader
CT	IslandUnpacker	gm2calo::IslandArtRecord
CB	HeaderUnpacker	gm2calo::AMCHeader gm2calo::CDFHeader
CH	CHUnpacker	gm2calo::QHistArtRecord
TTCR	FC7Unpacker	gm2ccc::CCCHeader gm2ccc::CCCArtRecord
CF	GpuFitResultUnpacker	gm2calo::FitResultArtRecord

## References

- [1] TRIUMF MIDAS homepage, accessed Jan 5, 2017. [https://midas.triumf.ca/MidasWiki/index.php/Main\\_Page](https://midas.triumf.ca/MidasWiki/index.php/Main_Page)