My Project

Generated by Doxygen 1.14.0

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

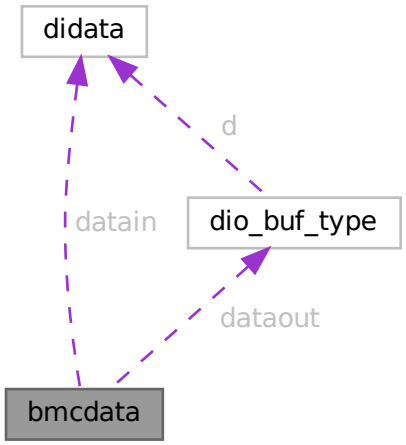
Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 bmcdata Struct Reference

Collaboration diagram for bmcdata:



Public Attributes

- double **pv_voltage**
- double **cc_voltage**
- double **input_voltage**
- double **b1_voltage**
- double **b2_voltage**
- double **system_voltage**
- double **logic_voltage**
- double **pv_current**
- double **cc_current**

- double **battery_current**
- struct didata **datain**
- union dio_buf_type **dataout**
- int32_t **adc_sample** [32]
- int32_t **dac_sample** [32]
- int32_t **utc**

The documentation for this struct was generated from the following file:

- daq.h

3.2 didata Struct Reference

Public Attributes

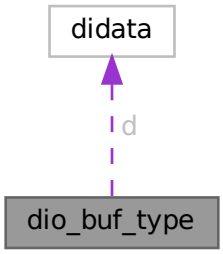
- uint32_t **D0**: 1
- uint32_t **D1**: 1
- uint32_t **D2**: 1
- uint32_t **D3**: 1
- uint32_t **D4**: 1
- uint32_t **D5**: 1
- uint32_t **D6**: 1
- uint32_t **D7**: 1

The documentation for this struct was generated from the following file:

- daq.h

3.3 dio_buf_type Union Reference

Collaboration diagram for dio_buf_type:



Public Attributes

- uint32_t **dio_buf**
- struct didata **d**

The documentation for this union was generated from the following file:

- daq.h

3.4 energy_type Struct Reference

Public Attributes

- volatile bool **once_gti**
- volatile bool **once_ac**
- volatile bool **iammeter**
- volatile bool **fm80**
- volatile bool **dumpload**
- volatile bool **homeassistant**
- volatile bool **once_gti_zero**
- volatile bool **comedi**
- volatile double **gti_low_adj**
- volatile double **ac_low_adj**
- volatile double **dl_excess_adj**
- volatile bool **ac_sw_on**
- volatile bool **gti_sw_on**
- volatile bool **ac_sw_status**
- volatile bool **gti_sw_status**
- volatile bool **solar_shutdown**
- volatile bool **solar_mode**
- volatile bool **startup**
- volatile bool **ac_mismatch**
- volatile bool **dc_mismatch**
- volatile bool **mode_mismatch**
- volatile bool **dl_excess**
- volatile uint32_t **speed_go**
- volatile uint32_t **im_delay**
- volatile uint32_t **im_display**
- volatile uint32_t **gti_delay**
- volatile uint32_t **sequence**
- volatile uint32_t **mqtt_count**
- volatile int32_t **rc**
- volatile int32_t **sane**
- volatile uint32_t **ten_sec_clock**
- volatile uint32_t **log_spam**
- volatile uint32_t **log_time_reset**
- pthread_mutex_t **ha_lock**
- volatile int16_t **di_16b**
- volatile int16_t **do_16b**
- double **adc** [16]
- double **dac** [16]
- MQTTClient **client_p**
- MQTTClient **client_sd**
- MQTTClient **client_ha**

The documentation for this struct was generated from the following file:

- bmc.h

3.5 ha_flag_type Struct Reference

Public Attributes

- volatile MQTTClient_deliveryToken **deliveredtoken**
- volatile MQTTClient_deliveryToken **receivedtoken**
- volatile bool **runner**
- volatile bool **rec_ok**
- int32_t **ha_id**
- volatile int32_t **var_update**
- volatile int32_t **energy_mode**

The documentation for this struct was generated from the following file:

- mqtt_rec.h

Chapter 4

File Documentation

4.1 bmc.h

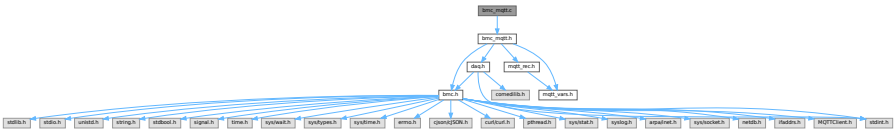
```
00001 /*
00002  * File:   bmc.h
00003  * Author: root
00004  *
00005  * Created on September 21, 2012, 12:54 PM
00006  */
00007
00008 #ifndef BMC_H
00009 #define BMC_H
00010
00011 #ifdef __cplusplus
00012 extern "C" {
00013 #endif
00014
00015 #include <stdlib.h>
00016 #include <stdio.h> /* for printf() */
00017 #include <unistd.h>
00018 #include <stdint.h>
00019 #include <string.h>
00020 #include <stdbool.h>
00021 #include <signal.h>
00022 #include <time.h>
00023 #include <sys/wait.h>
00024 #include <sys/types.h>
00025 #include <sys/time.h>
00026 #include <errno.h>
00027 #include <cjson/cJSON.h>
00028 #include <curl/curl.h>
00029 #include <pthread.h>
00030 #include <sys/stat.h>
00031 #include <syslog.h>
00032 #include <arpa/inet.h>
00033 #include <sys/socket.h>
00034 #include <netdb.h>
00035 #include <ifaddrs.h>
00036 #include "MQTTClient.h"
00037
00038 #define LOG_VERSION     "V0.02"
00039 #define MQTT_VERSION    "V3.11"
00040 #define TNAME  "maint9"
00041 #define LADDRESS        "tcp://127.0.0.1:1883"
00042 #ifdef __amd64
00043 #define ADDRESS         "tcp://10.1.1.172:1883"
00044 #else
00045 #define ADDRESS         "tcp://10.1.1.172:1883"
00046 #endif
00047 #define CLIENTID1       "Energy_Mqtt_BMC1"
00048 #define CLIENTID2       "Energy_Mqtt_BMC2"
00049 #define CLIENTID3       "Energy_Mqtt_BMC3"
00050 #define TOPIC_P         "comedi/bmc/data/bmc"
00051 #define TOPIC_SPAM      "comedi/bmc/data/spam"
00052 #define TOPIC_PACA      "home-assistant/comedi/bmc"
00053 #define TOPIC_PACB      "mateq84/data/#"
00054 #define TOPIC_AI        "comedi/bmc/data/ai"
00055 #define TOPIC_AO        "comedi/bmc/data/ao"
00056 #define TOPIC_DI        "comedi/bmc/data/di"
00057 #define TOPIC_DO        "comedi/bmc/data/do"
```

```
00058 #define QOS             1
00059
00060 #define TOPIC_SS        "mateq84/data/solar" // receive data testing
00061
00062 #define TIMEOUT         10000L
00063 #define SPACING_USEC    500 * 1000
00064 #define USEC_SEC        1000000L
00065
00066 #define CMD_SEC         30
00067 #define TIME_SYNC_SEC   30
00068
00069 #define SBUF_SIZ        16  // short buffer string size
00070 #define RBUF_SIZ        82
00071 #define SYSLOG_SIZ      512
00072
00073 #define LOG_TO_FILE         "/public/bmc/bmc_comedi.log"
00074 #define LOG_TO_FILE_ALT     "/tmp/bmc_comedi.log"
00075
00076 #define MQTT_RECONN     3
00077 #define KAI             60
00078
00079         extern FILE* fout; // logging stream
00080         extern struct energy_type E;
00081
00082         struct energy_type {
00083                 volatile bool once_gti, once_ac, iammeter, fm80, dumpload, homeassistant,
      once_gti_zero, comedi;
00084                 volatile double gti_low_adj, ac_low_adj, dl_excess_adj;
00085                 volatile bool ac_sw_on, gti_sw_on, ac_sw_status, gti_sw_status, solar_shutdown,
      solar_mode, startup, ac_mismatch, dc_mismatch, mode_mismatch, dl_excess;
00086                 volatile uint32_t speed_go, im_delay, im_display, gti_delay, sequence, mqtt_count;
00087                 volatile int32_t rc, sane;
00088                 volatile uint32_t ten_sec_clock, log_spam, log_time_reset;
00089                 pthread_mutex_t ha_lock;
00090                 volatile int16_t di_16b, do_16b;
00091                 double adc[16], dac[16];
00092                 MQTTClient client_p, client_sd, client_ha;
00093         };
00094
00095         void led_lightshow(int);
00096
00097 #ifdef __cplusplus
00098 }
00099 #endif
00100
00101 #endif /* BMC_H */
00102
```

4.2 bmc_mqtt.c File Reference

`#include "bmc_mqtt.h"`

Include dependency graph for bmc_mqtt.c:



Functions

- void **showIP** (void)
- void **skeleton_daemon** (void)
- char * **log_time** (bool log)
- void **timer_callback** (int32_t signum)
- void connlost (void *context, char *cause)
- void **delivered** (void *context, MQTTClient_deliveryToken dt)
- void **bmc_mqtt_init** (void)
- int32_t **msgarrvd** (void *context, char *topicName, int topicLen, MQTTClient_message *message)
- void **mqtt_bmc_data** (MQTTClient client_p, const char *topic_p)
- void **comedi_push_mqtt** (void)

Variables

- struct itimerval new_timer
- struct itimerval **old_timer**
- time_t **rawtime**
- MQTTClient_connectOptions **conn_opts_p** = MQTTClient_connectOptions_initializer
- MQTTClient_connectOptions **conn_opts_sd** = MQTTClient_connectOptions_initializer
- MQTTClient_connectOptions **conn_opts_ha** = MQTTClient_connectOptions_initializer
- MQTTClient_message **pubmsg** = MQTTClient_message_initializer
- MQTTClient_deliveryToken **token**
- char **hname** [256]
- char * **hname_ptr** = hname
- size_t **hname_len** = 12
- struct ha_flag_type ha_flag_vars_ss

4.2.1 Function Documentation

4.2.1.1 connlost()

```
void connlost (
            void * context,
            char * cause)
```

trouble in River-city

4.2.2 Variable Documentation

4.2.2.1 ha_flag_vars_ss

```
struct ha_flag_type ha_flag_vars_ss
```

Initial value:

```
= {
    .runner = false,
    .receivedtoken = false,
    .deliveredtoken = false,
    .rec_ok = false,
    .ha_id = COMEDI_ID,
    .var_update = 0,
}
```

4.2.2.2 new_timer

```
struct itimerval new_timer
```

Initial value:

```
= {
    .it_value.tv_sec = CMD_SEC,
    .it_value.tv_usec = 0,
    .it_interval.tv_sec = CMD_SEC,
    .it_interval.tv_usec = 0,
}
```

4.3 bmc_mqtt.h

```
00001
00002 #ifndef BMC_MQTT_H
00003 #define BMC_MQTT_H
00004
00005 #ifdef __cplusplus
00006 extern "C" {
00007 #endif
00008
00009 #include "bmc.h"
00010 #include "daq.h"
00011 #include "mqtt_rec.h"
00012 #include "mqtt_vars.h"
00013
00014 #define MQTT_RETRY 10
00015
00016         extern struct ha_flag_type ha_flag_vars_ss;
00017
00018         void mqtt_bmc_data(MQTTClient, const char *);
00019         void delivered(void *, MQTTClient_deliveryToken);
00020         int32_t msgarrvd(void *, char *, int, MQTTClient_message *);
00021         void connlost(void *, char *);
00022         void showIP(void);
00023         void skeleton_daemon(void);
00024         void bmc_mqtt_init(void);
00025         char * log_time(bool);
00026
00027         void timer_callback(int32_t);
00028         void comedi_push_mqtt(void);
00029
00030 #ifdef __cplusplus
00031 }
00032 #endif
00033
00034 #endif /* BMC_MQTT_H */
00035
```

4.4 daq.h

```
00001 /*
00002  * File:   daq.h
00003  * Author: root
00004  *
00005  * Created on September 21, 2012, 6:49 PM
00006  */
00007
00008 #ifndef DAQ_H
00009 #define DAQ_H
00010
00011 #ifdef __cplusplus
00012 extern "C" {
00013 #endif
00014
00015 #define PVV_C   0
00016 #define CCV_C   1
00017 #define SYV_C   2
00018 #define B1V_C   3
00019 #define B2V_C   4
00020 #define INV_C   5
00021 #define VD5_C   7
00022 #define PVC_C   8
00023 #define CCC_C   9
00024 #define BAC_C   10
00025
00026 #define LPCHANC        16
00027
00028 #define JUST_BITS false
00029
00030 #include <stdint.h>
00031 #include <comedilib.h>
00032 #include "bmc.h"
00033
00034         struct didata {
00035                 uint32_t D0 : 1; //
00036                 uint32_t D1 : 1; //
00037                 uint32_t D2 : 1; //
00038                 uint32_t D3 : 1; //
00039                 uint32_t D4 : 1; //
00040                 uint32_t D5 : 1; //
00041                 uint32_t D6 : 1; //
00042                 uint32_t D7 : 1; //
```

```
00043         };
00044
00045         union dio_buf_type {
00046                 uint32_t dio_buf;
00047                 struct didata d;
00048         };
00049
00050         typedef struct bmcdata {
00051                 double pv_voltage, cc_voltage, input_voltage, b1_voltage, b2_voltage, system_voltage,
      logic_voltage;
00052                 double pv_current, cc_current, battery_current;
00053                 struct didata datain;
00054                 union dio_buf_type dataout;
00055                 int32_t adc_sample[32];
00056                 int32_t dac_sample[32];
00057                 int32_t utc;
00058         }
00059         bmctype;
00060
00061         extern volatile struct bmcdata bmc;
00062         extern struct didata datain;
00063         extern struct dodata dataout;
00064
00065         extern int maxdata_ai, ranges_ai, channels_ai;
00066         extern int maxdata_ao, ranges_ao, channels_ao;
00067         extern int maxdata_di, ranges_di, channels_di, datain_di;
00068         extern int maxdata_do, ranges_do, channels_do, datain_do;
00069         extern int maxdata_counter, ranges_counter, channels_counter, datain_counter;
00070
00071         int init_daq(double, double, int);
00072         int init_dac(double, double, int);
00073         int init_dio(void);
00074         int adc_range(double, double);
00075         int dac_range(double, double);
00076         double get_adc_volts(int);
00077         int set_dac_volts(int, double);
00078         int set_dac_raw(int, lsampl_t);
00079         int get_dio_bit(int);
00080         int put_dio_bit(int, int);
00081         int set_dio_input(int);
00082         int set_dio_output(int);
00083         int get_data_sample(void);
00084         double lp_filter(double, int, int);
00085 #ifdef __cplusplus
00086 }
00087 #endif
00088
00089 #endif /* DAQ_H */
00090
```

4.5 mqtt_rec.h

```
00001
00002
00003 #ifndef MQTT_REC_H
00004 #define MQTT_REC_H
00005
00006 #ifdef __cplusplus
00007 extern "C" {
00008 #endif
00009
00010 #include "mqtt_vars.h"
00011
00012 #define RDEV_SIZE        10
00013
00014 #define SLEEP_CODE      0
00015 #define FLOAT_CODE      1
00016         //#define DEBUG_REC
00017         //#define GET_DEBUG
00018
00019 #define MBMQTT  1024
00020
00021         enum mqtt_id {
00022                 P8055_ID,
00023                 FM80_ID,
00024                 DUMPLOAD_ID,
00025                 HA_ID,
00026                 COMEDI_ID,
00027                 LAST_MQTT_ID,
00028         };
00029
00030         struct ha_flag_type {
00031                 volatile MQTTClient_deliveryToken deliveredtoken, receivedtoken;
```

```
00032                 volatile bool runner, rec_ok;
00033                 int32_t ha_id;
00034                 volatile int32_t var_update, energy_mode;
00035         };
00036
00037         extern FILE* fout;
00038
00039         int32_t msgarrvd(void *, char *, int, MQTTClient_message *);
00040         void delivered(void *, MQTTClient_deliveryToken);
00041
00042         bool json_get_data(cJSON *, const char *, cJSON *, uint32_t);
00043         bool fm80_float(const bool set_bias);
00044         bool fm80_sleep(void);
00045
00046
00047 #ifdef __cplusplus
00048 }
00049 #endif
00050
00051 #endif /* MQTT_REC_H */
00052
```

4.6 mqtt_vars.h

```
00001
00002
00003 #ifndef MQTT_VARS_H
00004 #define MQTT_VARS_H
00005
00006 #ifdef __cplusplus
00007 extern "C" {
00008 #endif
00009
00010 #define HA_SW_DELAY     400000  // usecs
00011 #define TOKEN_DELAY     600
00012 #define GTI_TOKEN_DELAY 300
00013
00014 #define QOS             1
00015
00016         void mqtt_ha_switch(MQTTClient, const char *, const bool);
00017         void mqtt_ha_pid(MQTTClient, const char *);
00018         void mqtt_ha_shutdown(MQTTClient, const char *);
00019         bool mqtt_gti_power(MQTTClient, const char *, char *, uint32_t);
00020         bool mqtt_gti_time(MQTTClient, const char *, char *);
00021
00022
00023 #ifdef __cplusplus
00024 }
00025 #endif
00026
00027 #endif /* MQTT_VARS_H */
00028
```

Index