

## **Muon HV Bleeder Software**

All software is being developed in ~jlorenc/daq/HVBleeder/HVBleederController

The directory contains the following files:

HVBleederController.h  
HVBleederController.hxx  
HVBleederController.cxx

### List of defines contained in **HVBleederController.h**

#define DIS (unsigned short) 0x1  
#define PWR (unsigned short) 0x2  
#define EN (unsigned short) 0x4

#define SHUNT\_ON 1  
#define SHUNT\_OFF 0

/\* Number of ports on Muon HV Bleeder Relay VME controller \*/  
#define PORT\_NUM 6

/\* Bits for checking the address success \*/  
#define ID\_OK (unsigned int)0  
#define ID\_ERROR (unsigned int)1

### Define a struct called **HVBleederCSR**

```
typedef struct
{
    unsigned short BLEEDER; /* 0x0 Mu HV Bleeder Controller Base Address */
    unsigned short ID; /* 0x2 Mu HV Bleeder ID (from Altera chip */
    unsigned short PORT[6]; /* 0x4 to 0xe Port 0 to Port 5 */
} HVBleederCSR;
```

List of public accessor functions

```
unsigned int compareID();  
unsigned int startBleedAll();  
unsigned int stopBleedAll();  
int readDischarge(int);  
int readPower(int);  
int readEnable(int);  
unsigned int readAllDischarge();  
unsigned int readAllPower();  
unsigned int readAllEnable();  
inline unsigned short getBleeder();  
inline unsigned short readID();  
inline unsigned short readPort(int);  
inline void writePort(int, unsigned short);  
inline unsigned short readPort0( );  
inline void writePort0(unsigned short);  
inline unsigned short readPort1();  
inline void writePort1(unsigned short);  
inline unsigned short readPort2();  
inline void writePort2(unsigned short);  
inline unsigned short readPort3();  
inline void writePort3(unsigned short);  
inline unsigned short readPort4();  
inline void writePort4(unsigned short);  
inline unsigned short readPort5();  
inline void writePort5(unsigned short);
```

Description of public accessor functions

***unsigned int compareID():***

Input Parameters:     None  
Returns:                1 if ID's match  
                             0 if ID's do not match

Description:            The ID on the board is a constant value set by LPM\_CONSTANT inside the Altera chip. This value is currently set to 48. This function is meant to be a check to make sure that the board we're talking to is really a Muon HV Relay VME Controller Card.

***unsigned int startBleedAll():***

Input Parameters:     None  
Returns:                0 (return value has no real meaning)

Description:            This functions sets the Discharge value on all the ports to 1. All of the Bleeder boxes mounted to these ports will shunt HV to GND and bleed off excess voltage.

***unsigned int stopBleedAll():***

Input Parameters:     None  
Returns:                0 (return value has no real meaning)

Description:            This functions clears the Discharge value on all the ports setting it to 0. All of the Bleeder boxes mounted to these ports will disconnect HV and GND and stop bleeding off excess voltage.

***int readDischarge(int):***

Input Parameters:     *int* port  
                             Port number (between 0 and 5)

Returns:                1 or 0 (Discharge value)

Description:            This function reads the Discharge value for an individual port.

***int readPower(int):***

Input Parameters:     *int* port  
Port number (between 0 and 5)  
Returns:               1 or 0 (Power value)  
  
Description:           This function reads the Power value for an individual port.

***int readEnable(int):***

Input Parameters:     *int* port  
Port number (between 0 and 5)  
Returns:               1 or 0 (Enable value)  
  
Description:           This function reads the Enable value for an individual port.

***unsigned int readAllDischarge():***

Input Parameters:     None  
Returns:               1 or 0  
  
Description:           This function reads all the ports and checks their discharge value. It returns 1 if the discharge value for all the ports is 1. If even one of the ports returns a nonzero discharge value, then this function returns 0.

***unsigned int readAllPower();***

Input Parameters:     None  
Returns:               1 or 0  
  
Description:           This function reads all the ports and checks their power value. It returns 1 if the power value for all the ports is 1. If even one of the ports returns a nonzero Power value, then this function returns 0.

***unsigned int readAllEnable();***

Input Parameters:     None  
Returns:               1 or 0

Description: This function reads all the ports and checks their Enable value. It returns 1 if the enable value for all the ports is 1. If even one of the ports returns a nonzero Enable value, then this function returns 0.

***inline unsigned short getBleeder():***

Input Parameters: None  
Returns: ID of Muon HV Bleeder Relay VME Controller

Description: This function does a register level read. It accesses the struct at `A[3..1] == 0` (defined as `HVBleederController.BLEEDER`) and gets the ID.

***inline unsigned short readID()***

Input Parameters: None  
Returns: ID of Muon HV Bleeder Relay VME Controller

Description: This function does a register level read at `A[3..1] = 0x1` and returns the board ID stored at that address.

***inline unsigned short readPort(int)***

Input Parameters: *int* port  
Port number (between 0 and 5)  
Returns: Value of the Port # register (EN,PWR,DIS)

Description: This function does a register level read at the given port register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN, PWR, DIS).

***inline void writePort(int, unsigned short)***

Input Parameters: *int* port, *unsigned short* value  
Port number (between 0 and 5)  
Value that you want to write (1 or 0)  
Returns: Nothing

Description: This function does a register level write at the given port register and writes “value” to port #. Only Data line 0

matters so this effectively sets and clears the discharge value for this particular port.

***inline unsigned short readPort0()***

Input Parameters:     None

Returns:                Value of the Port 0 register (EN0, PWR0, DIS0)

Description:            This function does a register level read at the Port 0 register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN0, PWR0, DIS0)

***inline void writePort0(unsigned short):***

Input Parameters:     *unsigned short* value

Returns:                Nothing

Description:            This function does a register level write at Port 0 , writing “value” to port 0. The VME board ignores all data lines except Data Line 0 which sets and clears the Discharge value for port 0.

***inline unsigned short readPort1()***

Input Parameters:     None

Returns:                Value of the Port 1 register (EN1, PWR1, DIS1)

Description:            This function does a register level read at the Port 1 register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN1, PWR1, DIS1)

***inline void writePort1(unsigned short):***

Input Parameters:     *unsigned short* value

Returns:                Nothing

Description:            This function does a register level write at Port 1, writing “value” to port 1. The VME board ignores all data lines except Data Line 0 which sets and clears the Discharge value for port 1.

***inline unsigned short readPort2()***

Input Parameters:     None

Returns:                Value of the Port 2 register (EN2, PWR2, DIS2)

Description:            This function does a register level read at the Port 2 register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN2, PWR2, DIS2)

***inline void writePort2(unsigned short):***

Input Parameters:     *unsigned short* value

Returns:                Nothing

Description:            This function does a register level write at Port 2 , writing “value” to port 2. The VME board ignores all data lines except Data Line 0 which sets and clears the Discharge value for port 2.

***inline unsigned short readPort3()***

Input Parameters:     None

Returns:                Value of the Port 3 register (EN3, PWR3, DIS3)

Description:            This function does a register level read at the Port 3 register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN3, PWR3, DIS3)

***inline void writePort3(unsigned short):***

Input Parameters:     *unsigned short* value

Returns:                Nothing

Description:            This function does a register level write at Port 3 , writing “value” to port 3. The VME board ignores all data lines except Data Line 0 which sets and clears the Discharge value for port 3.

***inline unsigned short readPort4()***

Input Parameters: None  
Returns: Value of the Port 4 register (EN4, PWR4, DIS4)  
  
Description: This function does a register level read at the Port 4 register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN4, PWR4, DIS4)

***inline void writePort4(unsigned short):***

Input Parameters: *unsigned short* value  
Returns: Nothing  
  
Description: This function does a register level write at Port 4 , writing “value” to port 4. The VME board ignores all data lines except Data Line 0 which sets and clears the Discharge value for port 4.

***inline unsigned short readPort5()***

Input Parameters: None  
Returns: Value of the Port 5 register (EN5, PWR5, DIS5)  
  
Description: This function does a register level read at the Port 5 register and returns the value of the register. The order of the 3 bits returned from most significant to least significant is (EN5, PWR5, DIS5)

***inline void writePort5(unsigned short):***

Input Parameters: *unsigned short* value  
Returns: Nothing  
  
Description: This function does a register level write at Port 5 , writing “value” to port 5. The VME board ignores all data lines except Data Line 0 which sets and clears the Discharge value for port 5.