ADDENDUM FOR THE

(GAB)
GENERIC ARCNET BOARD MANUAL

Brookhaven National Laboratory
Version 1.0

BY Jack Fried (email jfried@inst.inst.bnl.gov)
Web page (http://www.inst.bnl.gov/~jfried)

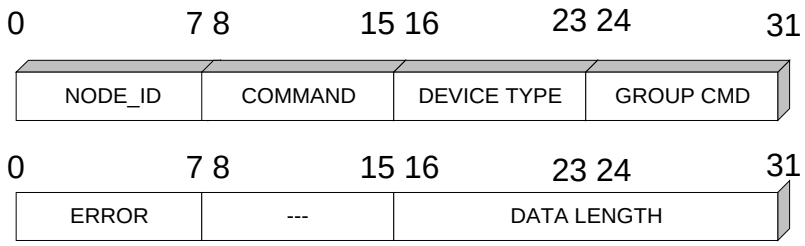
INDEX

Extra INFO

- **GAB test board**

Default power up function

The Default power up function is a new feature that allows the GAB to run a default function ~30 seconds after power up. This allows the GAB user to know the current state of FEM. The default function is located in the user program allowing it to be changed on the fly. If this feature is not required a user can ignore it and continue as if it does not exist.



How it works

The boot program will call the user program after 30 sec as if a packet was received over ARCNET. The GAB header will have the command parameter set at 0x0a, which is defined in the GFEM.H file as DEF_VAL. From this point the user program follows its normal flow and executes the command.

Example

```
Main()
{
  SETUP_PCKT;
  call_f();
}
void call_f()
{
        switch(pckt.cmd)
        {
           case   0x0a     /* load default values  */
                break;
                        .
                        .
                        .
           case  0x20     /*write port */
                break;
        }
}
```

If this function is not required then just ignore the function call.

Polling Function calls

This feature allows the GAB user to enter the user functions **without** ARCNET intervention. The Boot code allows the user to select the rate at which the boot code should call the function (from 1 to 256 seconds). This feature allows the GAB to be used as a voltage monitor or for any other applications that requires monitoring. Example: the TEC can monitor the G-LINK status bit every ~(1-10) secs for a lock

The Polling Function uses 2 registers shared by the Boot Code and the user program. The registers used are GUF[2] and GUF[3]. These registers are defined in the GFEM.H file.

DEFINITION OF REGISTERS

GUF[2]	FUNCTION CONTROL 0 = POLLING FUNCTION OFF 1= POLLING FUNCTION ON
GUF[3]	TIME DELAY IN SECONDS 1 = 1 second 2 = 2 seconds . . 30 = 30 seconds
GUF[4]	USER DEFINED (set to zero at power up or reset) (used to pass data from previous function call)

USAGE

```
void call_f()
{
        switch(pckt.cmd)
        {
           case   0x0a    /*    Load default values and  or polling function/
          {
                if(GUF[4] = 0)  /* load default */
                {
                   load default values
                  GUF[4]  = 1;
                 /* not setting GUF[2] or GUF[3]  polling function not enabled/
                }
                else
                {
                  /*check GUF[4] or set to new value other than 0 */
```

```
            do what ever function is necessary
          /*set polling/
          GUF[2] = 1;  /* turn on polling */
          GUF[3] = 10;  /* set  for ten seconds */
          }

     }
     break;
                    .
                    .
     case  0x20    /* turn on polling/
          {
                    GUF[2] = 1; /* turn on polling*/
                    GUF[3] = 5;  /* set for  5 seconds */
          }
     break;

     case  0x21    /* turn off polling or power up function/
          {
                    GUF[2] = 0;
          }
     break;

}
```

The boot program will call function 0x0a when the timer has elapsed. The boot code will also disabled GUF[2] and GUF[3] (set them to zero) when the function is called. The user function should reset GUF[2] and GUF[3] to desired values to ensure the function is called again.

GUF[4] was added so that the state of the program can be followed at any time. At power up GUF[4] is set to zero to indicate a power up state. After that the user can set it to any value he or she desires to control the flow of the program. One use for this register can be a counter to set the number of times a function should be entered. Another example: assuming a user function set a FEM board in some state that requires the GAB to wait 40 seconds for data. Now we can start the function set the timer to 40 seconds and set GUF[4] to a value that represents what operation the FEM is performing then return to the boot code to await the next command. After the 40 seconds the command 0x0a is received and looks at GUF[4] to complete the first operation.

User selectable group id

This feature allows the GAB user to group t FEMs in to any order a user might want. All that is required for the user to do, is to set GUF[5] to any value except the ones shown bellow. To call the group just set the group id in the GAB header to the value set in GUF[5] and all the nodes with matching values will receive the command.

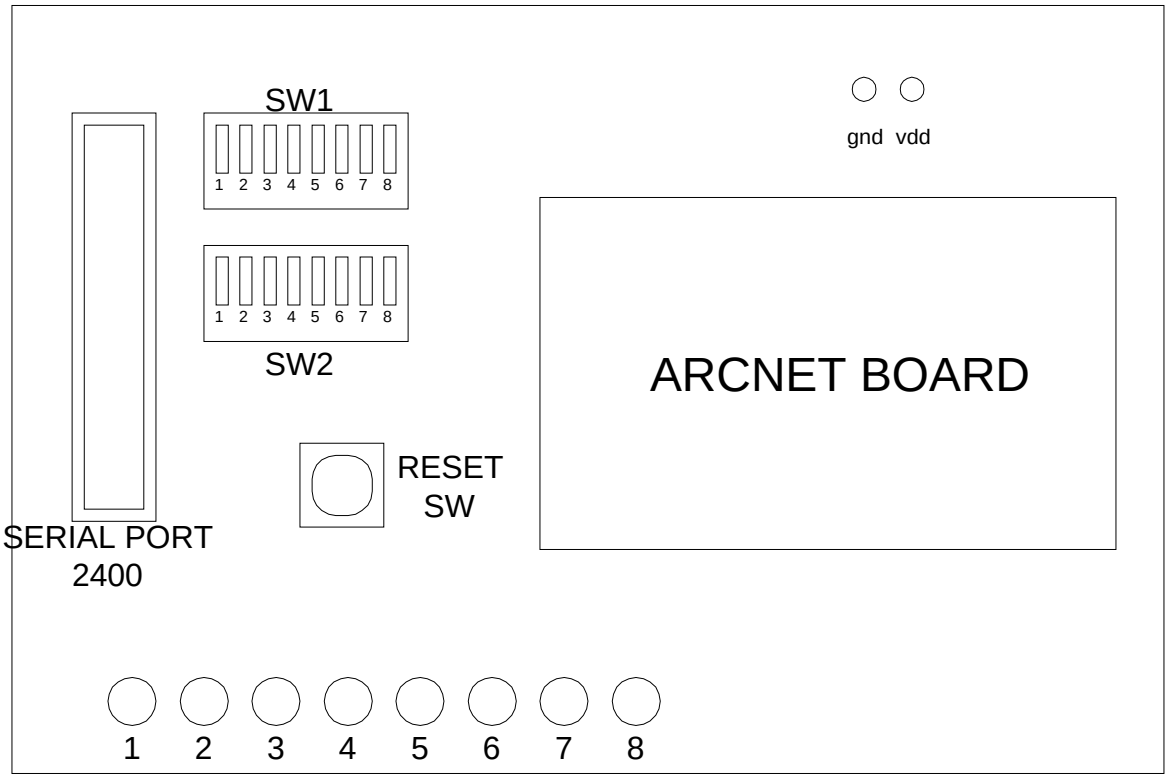
GUF[5] = 0 /* default */ (group id = first nibble of node id)

The values listed are NOT valid for GUF[5] (HEX)

0,1,10,20,30,40,50,60,70,80,90,a0,b0,c0,d0,e0,f0

any other values can be used in GUF[5].

GAB TEST BOARD



SW1 = MODE SELECT (DO NOT ADJUST)

1) ALL SWITCHES MUST BE SET TO OPEN

SW2 = NODE ID & IO PORT (1 = MSB 8 =LSB)

1) SET TO NODE ID
2) SET to value wanted to be read in by GAB

SW3 = RESET SWITCH

SERIAL PORT SETTINGS

BAUD RATE = 2400Bps
DATA BITS = 8
PARITY = NONE
STOP BITS = 1

FOR TEST MODE SET NODE ID to 0x02 (will come on network as node 1)

TEST MODE LEDs

1= TEST PR BIT	LED on = OK
2=PORT P1 TEST	LED on = OK
3=TEST ADDRESS LINES	LED on = OK
4=TEST READ FROM GAB	LED on = OK
5= TEST ALL CS LINES	LED on = OK
6= CHECK PORT R/W	LED on = OK
7= NETWORK TEST	LED on = OK
8= TEST WRITE FROM GAB	LED on = OK

WRITING TO THE LEDS

```
/*set register on test card*/
(* (char xdata *) (0xfef9 )) = 0x00;     /* You need to set the test board so that you can write to the leds*/

(* (char xdata * ) (0xfefa )) = 0x77;   /*  Write to the led*/
P3.3=1;              /*clock in the value to the leds latch */
P3.3=0;
```

READING PORT

num = (* (char xdata *) 0xfeff); reads the value on switch 2 and stores it in num