```
// C Code to control the DC Relay Module

// Relays are active low (i.e. writing a 0 to the relay turns it on).
#define RELAY_ON                   0
#define RELAY_OFF                  1

_Q void Control_DC_Relay ( uchar module_number, uchar relay_num, uchar
state )
// Sets the relay number to the appropriate state (on or off).
// Valid relay numbers are 0-2. Valid module numbers are 0-7.
{
  EXTENDED_ADDR module_addr;
  module_addr.sixteen_bit.page16 = module_number;
  module_addr.sixteen_bit.addr16 = RELAY_CONTROL_REGISTER;
  if(state) // turn relay off
  {
    state = state << relay_num;
    SetBits( state, module_addr.addr32 );
  }
    else // turn relay on
  {
    state = 1 << relay_num;
    ClearBits ( state, module_addr.addr32 );
  }
}

Q uchar Read_DC_Relay_Status ( uchar module_number )
// Reads the current state of the DC Relays. Valid module numbers are 0-7.
// Returns a character whose three least significant bits represents the
// three relays. For example, if 1 is returned (001 in binary), then Relay 0
// is off and the other relays are on. If 6 is returned (110 in binary),
// then relays 1 and 2 are off and 0 is on. The five most significant bits
// do not matter.
{
    EXTENDED_ADDR module_addr;
    uchar dc_relay_status;

    module_addr.sixteen_bit.page16 = module_number;
    module_addr.sixteen_bit.addr16 = RELAY_CONTROL_REGISTER;

    dc_relay_status = (uchar) FetchChar( module_addr.addr32 );

    return( dc_relay_status );
}
```