

How to directly write to/read from DCAN registers:

For accessing lower or upper 16-bits of a register, use HWREGH. (Have to add “2” to access upper 16-bits when dealing with byte peripherals such as CAN).

```
(1)  HWREGH(ui32Base + CAN_O_IF1ARB) = 0x4321;    // Write 0x4321 to LOWER 16-bit
(2)  HWREGH(ui32Base + CAN_O_IF1ARB + 2) = 0x8765; // Write 0x8765 to UPPER 16-bit
```

After executing the above 2 lines, CAN_0_IF1ARB should contain 0x87654321.

```
(3)  HWREGH(CANA_BASE + CAN_O_IF1ARB + 2) = 0x87650000 >> 16; // The >>16 is needed only when
you use 87650000 instead of 8765. IOW, are (2) and (3) effectively the same. You need the
>>16 because HWREGH is a 16-bit access and so the compiler will only write the lower 16 bits
of 0x87650000 to it.
```

For all 32-bit access, use HWREG_BP.

```
(4)  HWREG_BP(ui32Base + CAN_O_IF1ARB) = 0x87654321;    // Write 0x87654321 to the 32 bits
```

After executing the above line, CAN_0_IF1ARB should contain 0x87654321.

You can use HWREGB on the CAN registers to do an 8-bit access. You can see where it is used in *CAN_writeDataReg()* in the driverlib *can.h* file.