

Hi Mark,

I applied your modifications and it has improved things, I seem to be reading the correct number of bytes now, but I'm still got the following problems.

1. Single byte reads are still broken.
2. Reads of the PCF8575 chip, triggered from another task cause the IIC bus to lock up.

My start up routine looks like this

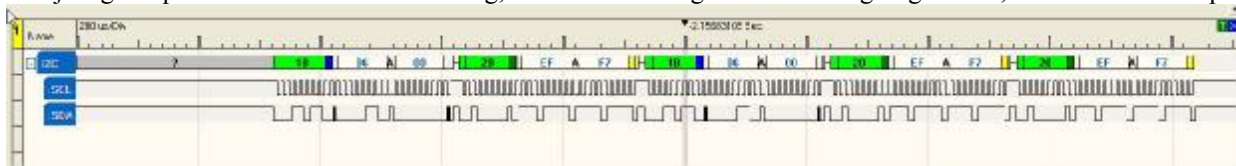
```
static const unsigned char IIC_WrTC665_FanSlow[] = {0x36, 0x06, 0x00}; // {address, register, data}
static const unsigned char IIC_WrTC665_FanFast[] = {0x36, 0x06, 0x0f}; // {address, register, data}
static const unsigned char IIC_WrTC665_FanSpeed[] = {0x36, 0x07}; // {address, register, data}
static const unsigned char IIC_RdTC665_FanSpeed[] = {2, 0x37, OWN_TASK}; // {address, register, data}
static const unsigned char IIC_Rd8575[] = {2, 0x41, OWN_TASK}; // {bytes, address, TASK_TO_WAKE}
static const unsigned char IIC_Rd8575_1[] = {1, 0x41, OWN_TASK}; // {bytes, address, TASK_TO_WAKE}

static void fnConfigIIC_Interface(void)
{
    IICTABLE tIICParameters;

    tIICParameters.Channel = 0;
    tIICParameters.usSpeed = 100; // 100k
    tIICParameters.Rx_tx_sizes.TxQueueSize = 128; // transmit queue size
    tIICParameters.Rx_tx_sizes.RxQueueSize = 128; // receive queue size
    tIICParameters.Task_to_wake = 0; // no wake on transmission

    if ((IICPortID = fnOpen( TYPE_IIC, FOR_I_O, &tIICParameters)) != 0) { // open the channel with defined configurations
        fnWrite(IICPortID, (unsigned char *)&IIC_WrTC665_FanSlow, sizeof(IIC_WrTC665_FanSlow)); // set the fan speed
        // fnRead(IICPortID, (unsigned char *)&IIC_Rd8575_1, 0); // start the read process of 16 bytes
        fnRead(IICPortID, (unsigned char *)&IIC_Rd8575, 0); // start the read process of 2 bytes
        fnWrite(IICPortID, (unsigned char *)&IIC_WrTC665_FanSlow, sizeof(IIC_WrTC665_FanSlow)); // set the fan speed
        fnRead(IICPortID, (unsigned char *)&IIC_Rd8575, 0); // start the read process of 2 bytes
        fnRead(IICPortID, (unsigned char *)&IIC_Rd8575, 0); // start the read process of 2 bytes
    }
}
```

I've just got repeated reads in there for testing, with the above I get the following Logic trace, which is what I expect.



If uncomment the line `// fnRead(IICPortID, (unsigned char *)&IIC_Rd8575_1, 0);` // start the read process of 16 bytes I get the following logic trace, which seems to lock up at the point a single byte read is performed.



Now ignoring single byte reads, because I don't need them I can do two byte reads and ignore the second byte. I still have problems. It seems I can put as many reads of address 0x20 as I want (which is my PCF8575 I/O chip) in the start up routine. Elsewhere in my code I have a task which looks at two switches and will either write an IIC message to speed up the fan or read the PCF8575. Writing to the fan works fine, but as soon as I press the switch that triggers a PCF8575 read I get the following on the bus and the IIC is locked up.

