TOTALFLOW

XSeries CIM Communication

Totalflow Technical Bulletin
Version 1.0 Revision AA (23 June 2010)

1 Introduction

The purpose of this Technical Bulletin is to describe the TFIO Communications Interface Module (CIM) when used with XSeries equipment.

XFC^{G4} (2102838 elex. board)

OR

XRC^{G4} (2103022 elex. board)

1.1 Background

The XSeries flow computers employ an I^{2}C bus for communication. This on-board communication bus was developed by Philips Electronics in the early 1980s. Totalflow XSeries devices have extended the I^{2}C bus to communicate with the Totalflow TFIO modules and daughter cards. The TFIO CIM module is only one of several TFIO modules that can be attached to the extended I^{2}C bus. Other TFIO modules include AI (analog input) modules, DI (digital input) modules, AO (analog output) modules and others.

1.2 CIM Architecture

The XSeries I^{2}C bus has a bandwidth of approximately 100bits/second. The XFC^{G4} and XRC^{G4} devices support four individual I^{2}C buses multiplexed to this primary I^{2}C bus. Before reading or writing to any I^{2}C device the appropriate multiplexed bus must be selected.

Most I^{2}C devices do not generate interrupts. The devices are polled; generally at a rate of once per second. However, the Keypad and CIM modules do generate an interrupt when they have received new data. The Keypad and CIMs share a single interrupt line. When an interrupt is generated, the main processor must poll the Keypad and CIMs to determine the interrupting device.

Each CIM incorporates two 23-byte receive buffers and two 31-byte transmit buffers. Once the CIM asserting the interrupt has been identified, the system must determine which buffer requires attention.

These activities, identifying the appropriate I^{2}C bus, identifying the individual CIM (or Keypad) and servicing the appropriate buffer, require processor time and bandwidth. This leads to inherent limitations that must be recognized by the User.

2 CIM Limitations

Due to the bandwidth limitations of the I^{2}C technology, a User cannot add TFIO CIM modules to the bus without considering the following recommendations.

NOTE:

Totalflow recommendations with regard to CIM communication on the I^{2}C bus are as follows.

- **G3 (XFC 2100204 elex. board, XRC 2100355 elex. board)**

  Do NOT use CIMs for Slave (interrupting) applications (i.e. Modbus Slave, Therms Slave, etc.)

- **G4 (XFC^{G4} 2102838 elex. board, XRC^{G4} 2103022 elex. board)**

  Limit CIM Slaves to 2 CIM devices communicating at 9600 baud or less

Failing to adhere to these recommendations could lead to system communication problems. Should the User attempt to exceed these recommendations, please refer to the next section, "Issues to Consider if Exceeding Totalflow Recommendations".

2.1 Issues to Consider if Exceeding Totalflow Recommendations (above)

If the User should attempt to exceed the above Totalflow Recommendations communication problems could ensue. The following is a list of technical issues that should be considered if communication problems are encountered.

- If you are having a CIM communications problem, try removing one of the other TFIO modules. If this helps the problem, it would indicate a possible bandwidth limitation.

- Some protocols can be more forgiving than others with respect to bandwidth. If an ASCII protocol is being used, try RTU… or vice versa.

- Lowering the baud rate might be helpful.

- Lowering the polling rate or selecting smaller packets can improve performance when bandwidth is limited.

- AGA7 applications are interrupt driven by the pulse input This places heavier timing demands on the processor than AGA3 applications. If temporarily suspending the AGA7 application improves performance it would indicate a possible bandwidth limitation.

- High CPU loading can affect CIM performance. If CPU loading is relatively high (> 70%), you might temporarily turn off a couple of applications. If this improves performance it could be an indicator of excessive processor loading or bandwidth limitations. Refer to the "Resources" tab in PCCU32.

3 Technical Explanation of Limitations

At 9600 baud, the 23-byte CIM receive buffer takes about 23 milliseconds (ms) to fill. If the system cannot keep up, the buffer overflows and data is lost. The system must read the receive buffers of all CIM modules every 23ms to avoid loss of data.

When the overhead of servicing an I^2C bus interrupt is added to the general latency of the system's multi-tasking environment, the time required to read the 23-byte receive buffer is about 6-7ms.

Keep in mind that the CIMs are not the only devices on the bus. Receiving a data packet via the I^2C bus generally requires a system response. These response packets translate into CIM write operations that also consume bandwidth.

4 CIM Configuration and Tuning Considerations

The worst case scenario is to have all the CIMs on the I^2C bus tied to the same RS-485 bus while receiving packets larger than 23 bytes. This would cause all of the CIMs to see all of the RS-485 traffic at the same time. This would lead to every byte of data on the RS-485 bus, having to be read from each CIM on the primary I^2C bus. Furthermore, all of the CIMs would be interrupting the system simultaneously (creating interrupt collisions). While this scenario may be useful for testing, having multiple CIMs tied to the same RS-485 bus is not a typical situation.

COM ports are generally connected to separate buses. One COM port is assigned for Therms; another for XMVs; another for Tanks; and still another for radios.

When CIMs are connected to separate buses, the incoming interrupts are more asynchronous (random). This leads to fewer interrupt collisions and improves the ability to recover data (via retries).

The best situation is when all receive packets are less than 23 bytes. XMVs are a good example. When packets are less than 23 bytes, the interrupt timing requirements are somewhat relaxed because the CIM can buffer the entire response. Also, with Modbus host protocol the bus timing can be controlled via request/response delays to reduce interrupt collisions.

However, even when you control bus timing and keep packets under 23 bytes, you may eventually exceed the bandwidth of the I^2C bus. Avoid loading the I^2C bus beyond 80%.