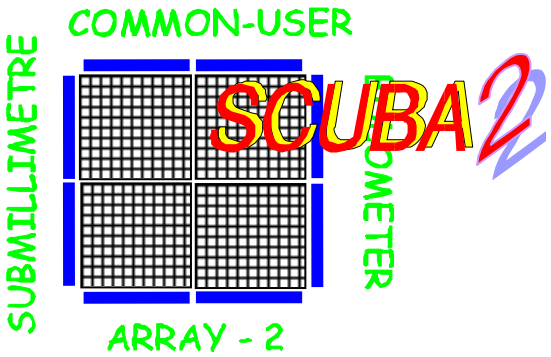

SCUBA-2 FTS Project Office
University of Lethbridge
Physics Department
4401 University Drive
Lethbridge, Alberta
CANADA
T1K 3M4

Tel: 1-403-329-2771
Fax: 1-403-329-2057
Email: brad.gom@uleth.ca
WWW: http://research.uleth.ca/scuba2/

Document Title: FTS-2 to RTS ICD

Document Number: SC2/FTS/SYS/005

Issue: Version 2.0

Date: 3 November 2006

Document Prepared By:	B. Zhang FTS-2 Software Engineer	Signature and Date:	03/11/06
Document Approved By:	D. A. Naylor FTS-2 Project Lead	Signature and Date:	03/11/06
Document Released By:	J. Molnar Canadian Project Manager	Signature and Date:	03/11/06

Change Record

Issue	Date	Section(s) Affected	Description of Change / Change Request Reference / Remarks
0.1	04/05/05	All	First draft
1.0	20/06/05	All	PDR version
1.1	05/07/05	2.1,2.2	Reworded the handshaking sequence description
2.0	3/11/06	All	CDR version. Removed old parallel port software, added PMC-485 details.

Applicable and Referenced Documents

Document Number	*Title*	*Number & Issue*
SC2/FTS/SOF/002	FTS-2 to OCS ICD	2.0
SC2/FTS/SYS/007	FTS-2 to JCMT ICD	2.0
SC2/SOF/S200/026	SCUBA-2 FTS and Polarimeter Coordination	1.0
	PMC-Parallel-485 NRC1 User Manual	A1
OCS/ICD/009	JAC Instrumentation Task (JIT) Library	
OCS/ICD/015	RTS DRAMA Client draft ICD	draft
RTS/XFG/004	Real-time Sequencer Hardware Design	
RTS/CAW/001	JCMT Real Time Sequencer (RTS) Client Interface Description	
SC2/SOF/S200/033	SCUBA-2 RTS Client Software Requirements – Part 1	
RTS/BDK/001.5/2002.03	Real-Time Sequencer Functional Requirements	

Contents

1. Introduction

The FTS-2 system is integrated into the JCMT Observatory Control System (OCS) and Real Time Sequencer (RTS) system through a control PC that acts as a Real Time Sequencer Client (RTSC). The FTS-2 control system, besides controlling various other mechanisms, must communicate with the moving mirror translation stage motion controller to obey OCS commands and also provide the mirror positions synchronized with the SCUBA-2 frames via the RTS. Synchronization between the RTSC and the motion controller is essential in order to ensure proper registration of the mirror position with the interferogram frames.

Since every RTS device must use the same hardware interface and perform many of the same functions, the JCMT uses a generalized software interface for any new RTS compliant instrument. With this interface, a new RTS compliant instrument only needs to provide 6 event-driven instrument-specific call-back procedures. This significantly simplifies the control software development for new RTS Client instruments.

1.1. Interface definition

This document defines the interfaces between the JAC Real Time Sequencer system (RTS) and the Fourier Transform Spectrometer (FTS-2) for the SCUBA-2 instrument. FTS-2 will be a fully compliant RTS DRAMA client, and will use the **rtsDClient** software library provided by the JAC (see OCS/ICD/015). The FTS-2 task is a JIT task and conforms to:
http://www.jach.hawaii.edu/JACdocs/JCMT/OCS/ICD/009/jit.pdf.
These actions are discussed in detail in the FTS-2 to OCS Interface Control Document (SC2/FTS/SOF/002)

The details of the RTS hardware and signal interfaces are described in the following sections. All cabling between the RTS system and the FTS-2 control PC will be provided by the JAC.

1.2. Acronyms and Abbreviations

FTS	-	Fourier Transform Spectrometer
ICD	-	Interface Control Document
JAC	-	Joint Astronomy Centre
JCMT	-	James Clerk Maxwell Telescope
OPD	-	Optical Path Difference
RTS	-	Real Time Sequencer
RTSC	-	Real Time Sequencer Client
U of L	-	University of Lethbridge
ZPD	-	Zero Path Difference

2. RTS Synchronization

2.1. Signal Interface

RTS clients have direct hardware connections to the JCMT RTS (Real-Time Sequencer) via a three-wire protocol. These control lines allow SCUBA-2 subsystem RTS clients to be synchronized with the SCUBA-2 Data Acquisition (DA) system. The three synchronization signals are:

- Subsystem Control (SC): simultaneous output from RTS to all clients
- Subsystem Ready (SR): output from each client to RTS
- Data Valid (DV): simultaneous output from RTS to all clients

This three-wire protocol consists of a preliminary SC/SR handshake followed by a series of integrations. Initially, the RTS master takes SC high and then waits for all the subsystems to acknowledge by taking their SR high. Next, the RTS takes SC low to complete handshaking between the RTS and the subsystem. For each integration, the RTS first takes SC high and the subsystem will respond with a high SR signal. The RTS then takes DV high, and then low on the RTS hardware clock's signal, ending the integration. The FTS-2 position capture and SCUBA-2 frame readout are synchronized to this falling edge. The RTS then takes SC low and waits for all the subsystems to take their SR low confirming that they have finished that step in the sequence. A full description of the handshaking sequence is provided in the "SCUBA-2 FTS and Polarimeter Coordination" document (SC2/SOF/S200/026).

2.2. FTS-2 operation

The operation of the FTS-2 system can be conceptualized in 3 levels: observation, scan and frame. An observation means all the data from the start of an experiment to its end as scheduled by JCMT OCS. A scan consists of acquiring an interferogram cube while the moving mirror assembly moves over the required distance. A frame is a single step in the interferogram, consisting of one frame from the SCUBA-2 DA system, encoded with the current position of the moving mirror assembly.

The start and end time of an observation will be set by JCMT OCS scripting files. The RTS sequence number is used to determine the start and end of each scan. Before each observation, the RTSC sets an internal sequence counter (or frame number) to 0. When the frame number reaches the maximum frame number for the scan, the RTSC uses the MODBUS/TCP protocol over Ethernet to tell the Soloist to reset its frame number and prepare for next scan. During the scan, the RTS synchronizes each frame and the mirror position readout using the three-wire protocol described above.

2.3. RTS event sequence

In order to better understand the FTS-2 control software, the RTS signal sequence can be divided into four phases according to function: initialization, handshaking, integration, and last integration. The relationship between the events in these four phases is shown below.

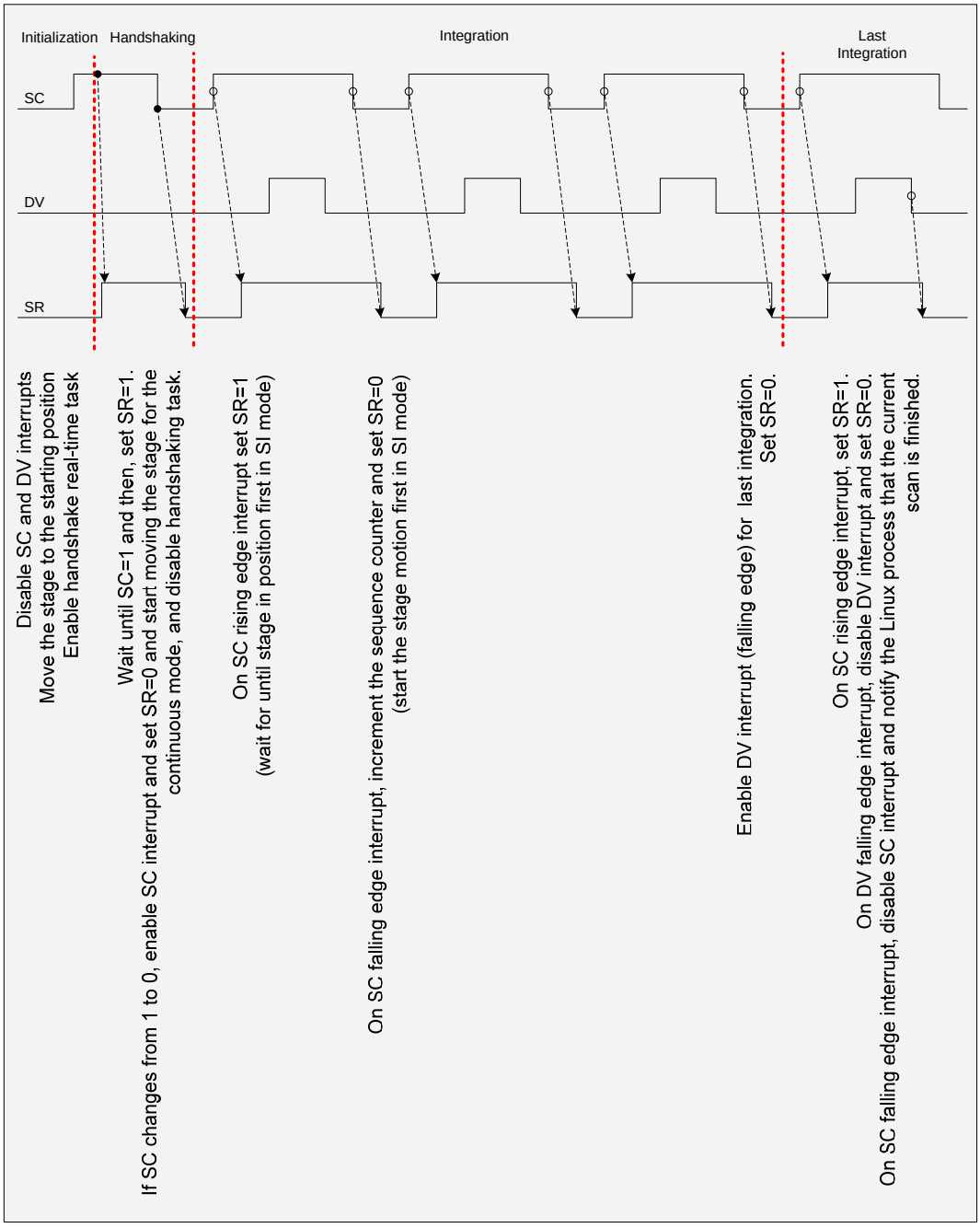

Figure 1. FTS-2 RTS event sequence

3. Hardware Connections

3.1. Connections between RTS, RTSC and Soloist

The RTS signal lines are differential RS485/422, therefore an interface card is required to interface the FTS-2 RTSC PC to the RTS system. A '[PMC-Parallel-485]' card was selected for this application. The FTS-2 RTSC accepts all 3 RTS lines via the PMC-Parallel-485 card and receives OCS DRAMA commands via Ethernet. The Soloist controller receives the DV line in order to trigger the position capture, and communicates with the FTS-2 RTSC via MODBUS/TCP over Ethernet. The PC will orchestrate the synchronization of OCS commands and replies, whereas the Soloist will perform the real-time synchronization of the position capture to the DA frame readout. A diagram of the electrical connections is given below.

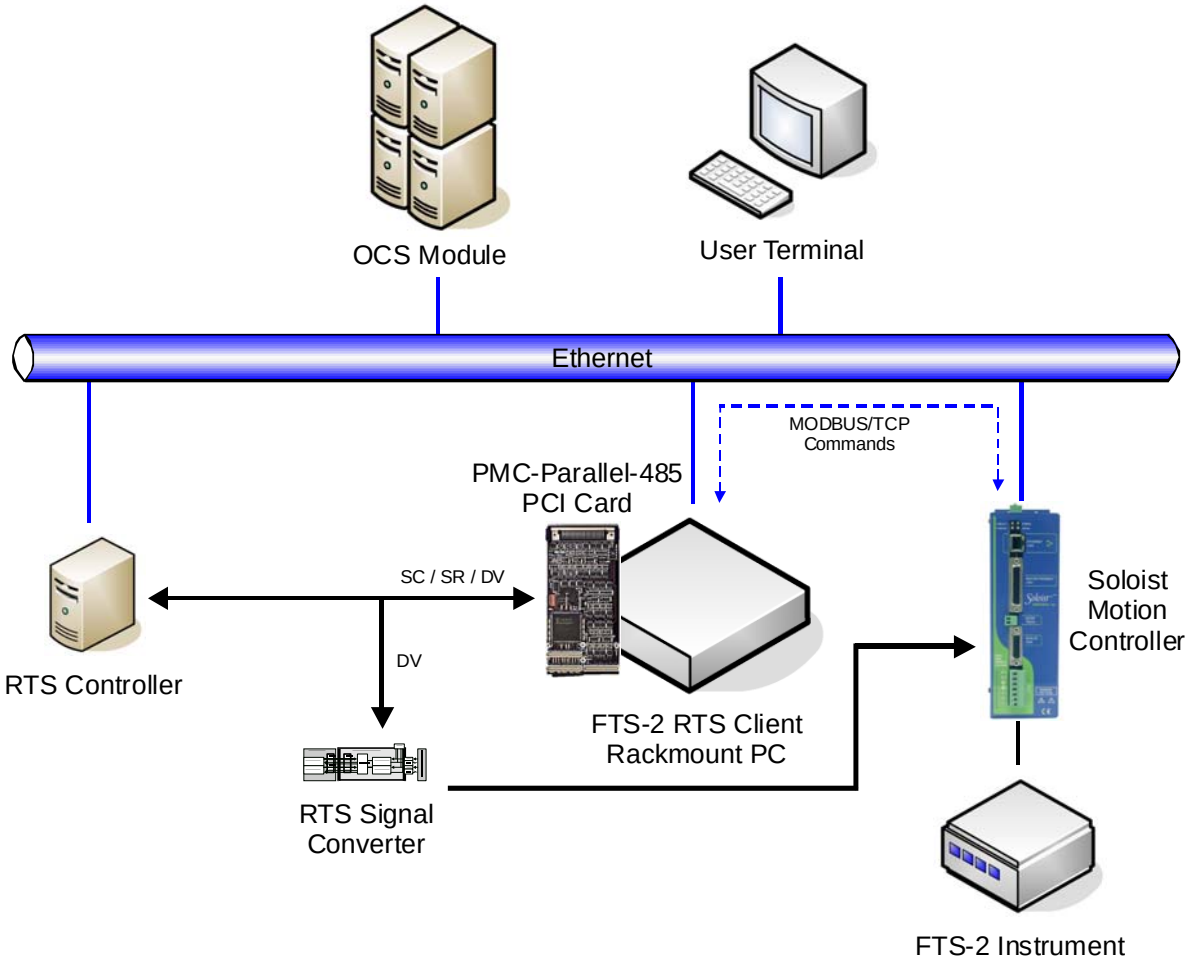

Figure 2. Schematic of the FTS-2 RTS Client electrical and network connections.

3.2. Description of hardware connections

The JAC will provide all cabling to the PMC card. The Soloist controller uses opto-isolated inputs, and a transceiver will be required in order for the Data Valid (DV) signal to drive the opto coupler input.

The PMC interface card accepts the differential RS-485 signals from the RTS system (SC, SR, and DV) with cabling and connectors provided by the JAC. DV is also connected to the position capture interrupt on the Soloist, after being converted to the proper voltage levels to drive the Soloist opto-isolator.

4. Control Software

The control software will be based on the **rtsDClient** DRAMA software library provided by the JAC (see OCS/ICD/015) .

Currently, the RTS software runs on a VxWorks computer mounted in a VMEBus chassis. Due mainly to budget constraints, the FTS-2 and POL-2 instruments will use the RTAI real-time Linux operating system instead of VxWorks for their RTS client software. At the time of writing however, the real-time Linux version of the RTS software interface was not yet available.

5. Prototype Control System

In order to prototype the FTS control system and test the performance of the Soloist communication system, an RTAI implementation of the RTS client software was written by the U of L. When the real-time Linux version of the RTS software interface is available from the JAC, this implementation will be modified to match the new interface.