



GRC DCSi-III

Users Guide

Revision: 1.0

Date: 04/17/08

Revisions

Rev.	Date	Author	Description
1.0	06/27/00	R. Zara	Initial version



Table of Contents

1. INTRODUCTION	3
2. ACRONYMS/DEFINITIONS	3
3. DESIGN OVERVIEW	4
4. HARDWARE	4
4. HARDWARE	5
4.1 DCSi MICROCONTROLLER BD	5
4.2 DCSi POWER SUPPLY/TERMINAL MODULE	5
4.3 DCSi CONNECTOR PIN ASSIGNMENTS	5
5. FUNCTIONAL OVERVIEW	6
5.1 CONFIGURATION	6
5.2 DIP SWITCH CONFIGURATION (SW1)	6
5.3 SOFTWARE	7
5.3.1 <i>Smart-CPU Recognition</i>	7
5.3.2 <i>DCSi Operating Modes</i>	7
5.3.3 <i>DCS Writes</i>	12
5.3.4 <i>Supported Modbus Functions</i>	13
5.3.5 <i>DCSi Slave Type</i>	13
5.3.6 <i>GRC Register Assignments</i>	13
5.3.7 <i>System Input Register</i>	16
5.3.8 <i>System Holding Register</i>	19



1. Introduction

This document defines the NWL Transformers Distributed Control Systems Interface (DCSi) Module for use with NWL's Graphic Rapper Controller. The DCSi Module will serve as an interface between NWL's Graphical Rapper Controller and the Plant Distributed Control Systems (DCS) using Modbus RTU or ASCII protocol.

2. Acronyms/Definitions

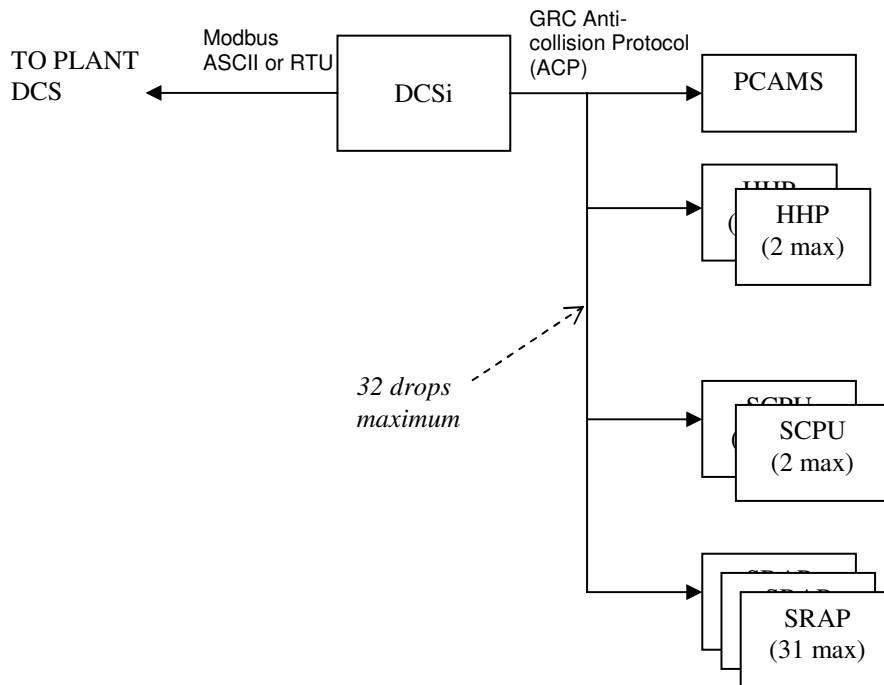
DCSi	Distributed Control System Interface – This is the product being designed. It will provide an interface between a plant's distributed control system Modbus link and the Graphical Rapper Controller communication link.
GRC	Graphical Rapper Controller – A control system that operates rappers and vibrators. The system consists of a card cage of boards. Each board is equipped with a microprocessor and can be individually communicated to.
SRAP	Smart-Rapper board – A board in the GRC card cage that energizes the rapping devices. It can either work independent of other boards in the card cage or be synchronized using an SCPU.
SCPU	Smart-CPU board – A board in the GRC card cage that synchronizes the operation of the SRAP boards.
HHP	Hand-held programmer – Hand-held user interface used to monitor and program the GRC boards.



3. Design Overview

The DCSi shall have two communication ports. Port 1 will communicate to the DCS system using either Modbus ASCII or Modbus RTU. Port 2 will communicate to the GRC system. The DCSi/GRC link addressing scheme allows for 1 DCSi, 1 PCAMS, 2 HHPs, 2 SCPUs, and 31 SRAPs. The communication link is limited to 32 of these devices simultaneously.

A system block diagram is shown below:



4. Hardware

4.1 DCSi Microcontroller Bd

NWL's Communications Microcontroller Board with the following features:

Processor	ST Micro UP3354, 40 MHz clock
Communication Ports	1 RS485 using microcontroller UART 1 RS232 using microcontroller UART
Memory	288K x 8 Flash 32K x 8 RAM
Watchdog Timer	

4.2 DCSi Power Supply/Terminal Module

NWL Transformers DCSi Power Supply/Terminal Module

Input	100-240 +10% -15% VAC, 50/60 Hz
Output	5 VDC \pm 5% @ 1000 mA
Environment	0° - 70° C operating, 5% - 95% relative humidity, non-condensing
Switches	10 position DIP switch for configuration, rear power switch
LED's	LED's showing link activity.

4.3 DCSi Connector Pin Assignments

The DCSi does not use hardware handshaking and only requires the TXD, RXD, and GND connections. *The Plant DCS may require signals to be tied to specific voltage levels. Check your documentation for more information.*

Sample cable from Computer to DCSi (note the twist of pins 2 & 3)

9 Pin PC (AT) RS232 Serial Port		<i>connects to</i>	Rs232 3-pin Terminal	DCSi, 9pin D-sub
Pin 1	Carrier Detect (DCD)			
Pin 2	Receive Data (RXD)	—————	Pin 2	Pin 3
Pin 3	Transmit Data (TXD)	—————	Pin 1	Pin 2
Pin 4	Data Terminal Ready (DTR)			
Pin 5	Ground (GND)	—————	Pin 3	Pin 5
Pin 6	Data Set Ready (DSR)			
Pin 7	Request to Send (RTS)			
Pin 8	Clear to Send (CTS)			
Pin 9	Ring Indicator (RI)			

RS485 Connector (J3) to ESP Power Optimizers

The Length, Type, and routing of RS485 cable can make or break a RS485 network. Please be sure to use RS485 certified cable, keep segment lengths to less than 4000 ft, and DO NOT route the cable along



with or next to high current devices/cabling or other noise emitting sources. Refer to NWL Drawing B117136 Rev A for NWL recommended RS485 guidelines.

J3 Pin #	Description
1	High
2	Low
3	Signal Common

For convenience also located on the front panel of the DCSi is a network port (RJ45) which can be used to attach a GVC Display module for viewing and/or confirmation of Optimizer network data. Power for the GVC is provided via this port therefore no remote power supply is required.

5. Functional Overview

5.1 Configuration

NWL Link	9600/19200 Baud (automatic detection) 8 data bits, 1 stop bit, no parity
DCS Link	19200/38400 Baud configurable Modbus RTU: 1 start bit, 8 data bits, parity and stop bits configurable Modbus ASCII: 1 start bit, 7 data bits, parity and stop bits configurable Modbus Slave Addresses configurable

5.2 DIP Switch Configuration (SW1)

The DCSi's configuration is controlled via a 10 position Dip switch located on the front of the unit. The following tables provide the details for each switch position.

SW1-1	DCS Link Baud Rate
Off	38400
On	19200 (default)

SW1-2	DCSi Mode
Off	Redundant
On	Primary (default)

SW1-3	# of NWL Nodes *
Off	1 (default)
On	2

* - For best performance SWI-3 should be set to scan the minimum number of nodes connected. Therefore if the attached rs485 communication line has 12 or fewer Optimizer nodes setting this switch to off will provide better performance.

SW1-4	Unused



SW1-5	SW1-6	Modbus Slave Address
On	On	1 (default)
On	Off	2
Off	On	3
Off	Off	4

SW1-7 Bit 6	Protocol
Off	Modbus ASCII
On	Modbus RTU (default)

SW1-8	Extended Registers *
Off	No
On	Yes (default)

* - Note: Extended Registers refers to the Fire Table Registers (Input Registers 30019 - 30048). Extended Registers are only available on a DCSi licensed for one GRC and firmware ver 1.06 or newer. If your application does not require this information, SW1-8 should be set to the off position to increase bandwidth and scanning rate of remaining registers.

SW1-9	Unused

SW1-10	Rs485 120 ohm terminating resistor
Off	None
On	In Circuit (default)

5.3 Software

The DCSi operates two (2) communications ports - 1 NWL Link, and 1 DCS Link. The NWL Link protocol and communication port parameters are automatically configured. The DCSi supports the Anti-collision protocol for the NWL Link.

5.3.1 Smart-CPU Recognition

The DCSi will scan the NWL Link after startup to determine the Smart-CPU's that are on the link. Smart-CPU's that do not respond will be re-checked for communications within every 30 seconds when operating in the *Operation Overview* mode (see below). A register will be available so that the DCS can read the communication status of all SCPU's in the system.

5.3.2 DCSi Operating Modes

To facilitate various monitoring requirements of the DCS, there are several monitoring modes that the DCSi performs. The DCS has the ability to command these modes via a write to the *DCSi operating mode* register (49998) in the DCSi. Only one mode can be operational at one time. They are described in the table below:



DCSi Operating Mode	Description
Operation Overview	Monitors holding registers and input registers of SCPUs at a fixed period.
Rapper firing monitor	Causes the DCSi to monitor the firing activity of a single SCPU. The DCS uses function code 20 to upload the firing data. The SCPU being monitored is defined in the <i>DCSi Operation Mode SCPU Address</i> register (49997).

5.3.2.1 Operation Overview

This operating mode is designed for the DCS to monitor the input registers and holding registers of **all** SCPUs on the link. The input registers will be refreshed at a maximum of once per minute. The Holding Registers will be refreshed once per 10 minutes maximum.

5.3.2.2 Rapper Firing Monitor

This operating mode allows the DCS to monitor the firing activity of one SCPU. The data is transferred in buffers with time-stamps so that the DCS can obtain an accurate record of the rapper firing. If the DCS requests this mode for an SCPU that is not communicating, the DCSi will first attempt to communicate to the SCPU and then retry every 30 seconds.

5.3.2.2.1 SCPU Buffer

The SCPU buffer is sized to keep 15 seconds worth of fire data. Since the SCPU can fire a device once every .25 seconds, the buffer will have 60 entries. Each entry shall be formatted as follows:

Name	Type	Description
wRap1	UWORD	1 st Rapper fired (1 – 1024)
wRap2	UWORD	2 nd Rapper fired (1 – 1024)
wRap3	UWORD	3 rd Rapper fired (1 – 1024)
wRap4	UWORD	4 th Rapper fired (1 – 1024)
wRap5	UWORD	5 th Rapper fired (1 – 1024)
wRap6	UWORD	6 th Rapper fired (1 – 1024)
LtimeStamp	LONG	Time stamp in .25 second increments. Zeroed when SCPU is powered-up. Will rollover once every 34 years.

The buffer is a circular queue. An entry is made when rappers are fired. An entry is removed when the DCSi requests data from the queue. The queue uses a put index, get index, and number of entries count. A handshake is used between the DCSi and the SCPU to keep track of the entries that are received by the DCSi.

Rules of the queue are as follows:

When the put index and the get index are equal and the number of entries count is zero, the queue is empty. When the put index and the get index are equal and the number of entries count is non-zero, the queue is full.

If the put index is greater than the get index, then the number of entries count equals the get index minus the put index.

If the get index is greater than the put index, then the number of entries count equals the length of the queue minus the get index plus the put index.

The queue is transmitted in a message with a structure that indicates the number of queue entries that are being transmitted. Each request of data must be acknowledged. Upon the acknowledgment, the queue's get



index is updated accordingly. This is to guarantee that the DCSi properly received the data. If another request is made prior to an acknowledgment, then the same data will be retrieved.

If the amount of data is greater than the maximum allowed message length, then the data must be transferred in packets. The message will indicate if the whole table, partial table, or end of table has been received. Since data is transferred in BCD, the long word needs to be transmitted in 5 bytes.

5.3.2.2.2 Rapper Fire Queue Upload DCSi/SCPU Message

This message is used to upload the rapper fire queue. The queue is uploaded in packets. To keep the master and slave devices in sync, a sequence number is used. The master sends a sequence number to start at the beginning of the queue, and increments the sequence number on subsequent requests. This is to guarantee that the master is receiving the correct data packet, and it acknowledges to the slave that the master received the packet. When the slave receives the acknowledgment, it adjusts its get index and number of entries count accordingly.

Master Command

Start Byte	0xFC or 0xFD	Hex
Destination Address		BCD
Source Address		BCD
Command	7	BCD
Sequence Number	See note 1	BCD
Data Endbyte	0xF0	Hex
Checksum	(calculated)	Hex
Transmit Endbyte	0xF1	Hex

Notes:

- Sequence number designates the part of the table being requested.
- 0=First Packet; restarts buffer download
- 1-95=Subsequent Packet Numbers; cycles through these numbers
- 96,97,98= Reserved (see slave response below)
- 99= Final packet acknowledge

Slave Response

Start Byte	0xFB	Hex
Destination Address		BCD
Source Address		BCD
Sequence Number	See note 1	BCD
Data	(49 bytes max)	BCD
.		.
.		.
.		.
Data		.
Data Endbyte	0xF0	Hex
Checksum	(calculated)	Hex
Transmit Endbyte	0xF2	Hex

Notes:

- Sequence number designates the part of the table being sent.
- 0=First Packet



1-95=Subsequent Packet Numbers; cycles through these numbers until everything is downloaded
 96= Beginning and end of data in packet
 97=End Packet
 98=Out of sequence
 99=Acknowledge

Example transactions:

Empty queue:

Master requests data with sequence equal to 0, slave responds with sequence 96 followed by endbyte to indicate there is no data, master does not need to respond with acknowledge.

Queue is less than one packet long:

Master requests data with sequence equal to 0. Slave responds with sequence 96 followed with data. Master sends request with sequence equal to 99 to indicate acknowledge. Slave responds with 99 and no data.

Queue is more than one packet long:

Master requests data with sequence equal to 0. Slave responds with sequence 0 followed with data. Master sends request with sequence equal to 1. Slave adjusts its get index and number of entries count and responds with sequence equal to 1. Master and slave continue in this fashion until slave finally responds with a 97 sequence number indicating that the end of the queue has been reached. Master sends request with sequence equal to 99 to indicate acknowledge. Slave responds with 99 and no data.

If master fails to receive data:

Master requests data with sequence equal to 0. Slave responds with sequence 0 followed with data. Master sends request with sequence equal to 1. Slave adjusts its get index and responds with sequence equal to 1. Master misses the reception, so it sends a request for 1 again. Slave resends the sequence 1 data. Master receives the data and sends a request for sequence 2. Slave adjusts its get index and sends the next packet. And so on...

5.3.2.2.3 DCSi Rapper fire queue

The DCSi will keep a similar queue. All data received from the SCPU gets stored into the DCSi's fire queue. This queue can be read by the DCS. The queue has its own put index, get index, and number of entries count. The queue also follows the same rules as stated for the SCPU's queue.

5.3.2.2.4 Rapper Fire Queue Upload Modbus link Message

For the Modbus link, similar handshaking shall be used for accessing fire queue. Function code 20 – Read General Reference – shall be used to upload the queue.

Function code 20 has the following format:

Request

0	1	2	3	4	5	6	7	8	9	10	11
ADDR	FUNC	BYTE CNT	REF TYPE	FILE NO HI	FILE NO LO	REQ ADDR HI	REQ ADDR LO	REG CNT HI	REG CNT LO	LRC or CRC HI	CRC LO



BYTE CNT – Total number of bytes excluding address, function code, byte count, and error check fields.

REF TYPE – Unused; set to 0

FILE NO – 0 indicates fire queue

REQ ADDR – 0

REG CNT – This is used as a handshake with the slave to guarantee that data is properly being received.

The master first sets REG CNT equal to 0. This indicates to the slave to start at the beginning of the queue.

When the master receives the first packet, it will increment the number to indicate that it wants the next packet. This will alert the slave that the master properly received the data so it will adjust its get index and entry count for the fire queue. If the master fails to get a response from the slave, it could be for one of two reasons. Either the slave never saw the request because the request got corrupted on its way to the slave, or the slave responded to the master, and the response got corrupted. In either case, the master can resend the same sequence number. If the slave already responded to the sequence number, it will recognize this and resend the packet. Packet numbers must be sequential. If the slave sees an out of sequence packet number, it will indicate this as an error in its response.

The following values are used:

0 = Start of queue

1 – 250 = Remaining packets (rolls over to 1 if queue continues past 250)

251-253 = *Reserved; see slave responses below*

254 = Acknowledge

Response

0	1	2	3	4	5	6	7	8	9								
ADDR	FUNC	BYTE CNT	REF TYPE	FILE NO HI	FILE NO LO	REQ ADDR HI	REQ ADDR LO	REG CNT HI	REG CNT LO								
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
FIRE QUEUE RECORD																LRC or CRC HI	CRC LO

BYTE CNT – Total number of bytes excluding address, function code, byte count, and error check fields.

REF TYPE – Echo of request

FILE NO – Echo of request

REQ ADDR – Echo of request

REG CNT – Used as a handshake with the master. The following values are used:

0=First Packet

1-250=Subsequent Packet Numbers; cycles through these numbers until everything is downloaded

251= Beginning and end of data in packet

252=End Packet

253=Out of sequence

254=Acknowledge

FIRE QUEUE RECORD – The fire queue record is described in section 0. Records are sent as complete entities. Partial records are not permitted. Up to 15 records can be sent in one response.

Example transactions:

Empty queue:

Master requests data with REG CNT equal to 0. Slave responds with REG CNT equal to 251 and no data. Master does not need to respond with acknowledge.



Queue is less than one packet long:

Master requests data with REG CNT equal to 0. Slave responds with REG CNT equal to 251 followed with record(s). Master sends request with REG CNT equal to 254 to indicate acknowledge. Slave responds with REG CNT equal to 254 and no data.

Queue is more than one packet long:

Master requests data with REG CNT equal to 0. Slave responds with REG CNT equal to 0 followed with records. Master sends request with REG CNT equal to 1. Slave adjusts its get index and number of entries count and responds with REG CNT equal to 1. Master and slave continue in this fashion until slave finally responds with a REG CNT equal to 252 indicating that the end of the queue has been reached. Master sends request with REG CNT equal to 254 to indicate acknowledge. Slave responds with REG CNT equal to 254 and no data.

If master fails to receive data:

Master requests data with REG CNT equal to 0. Slave responds with REG CNT equal to 0 followed with data. Master sends request with REG CNT equal to 1. Slave adjusts its get index and responds with REG CNT equal to 1. Master misses the reception, so it sends a request for 1 again. Slave resends the REG CNT 1 data. Master receives the data and sends a request for REG CNT 2. Slave adjusts its get index and sends the next packet. And so on...

If master receives an out of sequence REG CNT:

If the slave responds with REG CNT equal to 253, the master should respond with a REG CNT equal to 0 to force the slave to start at the first record in the queue.

5.3.3 DCS Writes

When the DCS writes a value to the SCPU via the DCSi, the DCSi should do the write followed by an immediate read. The value read should be updated in the SCPU registers internal to the DCSi. This way, the DCS can read back the value (after a certain amount of delay), to determine if the value was accepted or rejected. If it was accepted, then the value will have been changed. If rejected, then the value would be the same as previous.



5.3.4 Supported Modbus Functions

Function #	Description	Register Range
3	Read Holding Registers	40001 - 50000
4	Read Input Registers	30001 - 40000
6	Preset Single Register	40001 - 50000
16	Preset Multiple Registers	40001 - 50000
17	Report Slave ID	n/a
20	Read General Reference	N/a
21	Write General Reference	N/a

5.3.5 DCSi Slave Type

The DCSi will report a Slave ID (function code 17) of 3 for a **Micro 584** Slave Type. A Micro 584 device has the following restrictions:

Function #	Maximum Registers per Query	Maximum Registers per Response
3	125	125
4	125	125
6	1	1
16	100	100

5.3.6 GRC Register Assignments

5.3.6.1 Input Registers

The input registers are for monitoring only. Each SCPU is assigned a block of input register numbers as shown below:

Register # Range	SCPU Address
30001 – 30200	1
30201 – 30400	2

The function of each register is shown in the following table. The register numbers in the table are for SCPU address 1. By offsetting the register addresses in multiples of two hundred, other SCPUs are addressed.

Register #	Description	Valid Range	Units
30001	Software Version ID	0 – 9999	
30002	Average Opacity	0 – 999	% Opacity x 10
30003	Average Boiler Load	0 – 9999	
30004	S-Rap Alarms Exist	Bit 0, Bit 1, Bit 2	Bit 0: 1=at least one alarmed rapper exists Bit 1: 1=at least one SRAP cannot be communicated to via the I2C link



			Bit 2: 1=program off
30005	S-Raps 1 - 13 alarms	Bits 0 - 12	1 = alarm
30006	S-Raps 14 - 26 alarms	Bits 0 - 12	1 = alarm
30007	S-Raps 27 - 39 alarms	Bits 0 - 12	1 = alarm
30008	S-Raps 40 - 52 alarms	Bits 0 - 12	1 = alarm

Register #	Description	Valid Range	Units
30009	S-Raps 53 - 64 alarms	Bits 0 - 11	1 = alarm
30010	S-Raps 1 - 13 communication failure	Bits 0 - 12	1 = failure
30011	S-Raps 14 - 26 communication failure	Bits 0 - 12	1 = failure
30012	S-Raps 27 - 39 communication failure	Bits 0 - 12	1 = failure
30013	S-Raps 40 - 52 communication failure	Bits 0 - 12	1 = failure
30014	S-Raps 53 - 64 communication failure	Bits 0 - 11	1 = failure
30015	Active Operating Program	1-5	Prog #
30016	Operating/POR Program On/Off	0,1	Off, On
30017	Remote Software Select	0-2	0 = Disable, 1 = Discrete, 2 = Network
30018	Remote Enable State	0,1	Disabled, Enabled
30019	Fire Table: t-4 Device 1	0, 1 - 1024	Empty, Output #
30020	Fire Table: t-4 Device 2	0, 1 - 1024	Empty, Output #
30021	Fire Table: t-4 Device 3	0, 1 - 1024	Empty, Output #
30022	Fire Table: t-4 Device 4	0, 1 - 1024	Empty, Output #
30023	Fire Table: t-4 Device 5	0, 1 - 1024	Empty, Output #
30024	Fire Table: t-4 Device 6	0, 1 - 1024	Empty, Output #
30025	Fire Table: t-3 Device 1	0, 1 - 1024	Empty, Output #
30026	Fire Table: t-3 Device 2	0, 1 - 1024	Empty, Output #
30027	Fire Table: t-3 Device 3	0, 1 - 1024	Empty, Output #
30028	Fire Table: t-3 Device 4	0, 1 - 1024	Empty, Output #
30029	Fire Table: t-3 Device 5	0, 1 - 1024	Empty, Output #
30030	Fire Table: t-3 Device 6	0, 1 - 1024	Empty, Output #
30031	Fire Table: t-2 Device 1	0, 1 - 1024	Empty, Output #
30032	Fire Table: t-2 Device 2	0, 1 - 1024	Empty, Output #
30033	Fire Table: t-2 Device 3	0, 1 - 1024	Empty, Output #
30034	Fire Table: t-2 Device 4	0, 1 - 1024	Empty, Output #
30035	Fire Table: t-2 Device 5	0, 1 - 1024	Empty, Output #
30036	Fire Table: t-2 Device 6	0, 1 - 1024	Empty, Output #
30037	Fire Table: t-1 Device 1	0, 1 - 1024	Empty, Output #
30038	Fire Table: t-1 Device 2	0, 1 - 1024	Empty, Output #
30039	Fire Table: t-1 Device 3	0, 1 - 1024	Empty, Output #
30040	Fire Table: t-1 Device 4	0, 1 - 1024	Empty, Output #
30041	Fire Table: t-1 Device 5	0, 1 - 1024	Empty, Output #
30042	Fire Table: t-1 Device 6	0, 1 - 1024	Empty, Output #
30043	Fire Table: t-0 Device 1	0, 1 - 1024	Empty, Output #
30044	Fire Table: t-0 Device 2	0, 1 - 1024	Empty, Output #
30045	Fire Table: t-0 Device 3	0, 1 - 1024	Empty, Output #
30046	Fire Table: t-0 Device 4	0, 1 - 1024	Empty, Output #
30047	Fire Table: t-0 Device 5	0, 1 - 1024	Empty, Output #



30048	Fire Table: t-0 Device 6	0, 1 – 1024	Empty, Output #
30049	Output Status: Devices 1-16 (SRAP1 – bit encoded)	Bits 0 – 15	1 = Alarmed
30050	Output Status: Devices 17-32 (SRAP2 – bit encoded)	Bits 0 – 15	1 = Alarmed
30051	Output Status: Devices 33-48 (SRAP3 – bit encoded)	Bits 0 – 15	1 = Alarmed
30052	Output Status: Devices 49-64 (SRAP4 – bit encoded)	Bits 0 – 15	1 = Alarmed
30053	Output Status: Devices 65-80 (SRAP5 – bit encoded)	Bits 0 – 15	1 = Alarmed
30054	Output Status: Devices 81-96 (SRAP6 – bit encoded)	Bits 0 – 15	1 = Alarmed
30055	Output Status: Devices 97-112 (SRAP7 – bit encoded)	Bits 0 – 15	1 = Alarmed
30056	Output Status: Devices 113-128 (SRAP8 – bit encoded)	Bits 0 – 15	1 = Alarmed
30057	Output Status: Devices 129-144 (SRAP9 – bit encoded)	Bits 0 – 15	1 = Alarmed
30058	Output Status: Devices 145-160 (SRAP10 – bit encoded)	Bits 0 – 15	1 = Alarmed
30059	Output Status: Devices 161-176 (SRAP11 – bit encoded)	Bits 0 – 15	1 = Alarmed
30060	Output Status: Devices 177-192 (SRAP12 – bit encoded)	Bits 0 – 15	1 = Alarmed
30061	Output Status: Devices 193-208 (SRAP13 – bit encoded)	Bits 0 – 15	1 = Alarmed
30062	Output Status: Devices 209-224 (SRAP14 – bit encoded)	Bits 0 – 15	1 = Alarmed
30063	Output Status: Devices 225-240 (SRAP15 – bit encoded)	Bits 0 – 15	1 = Alarmed
30064	Output Status: Devices 241-256 (SRAP16 – bit encoded)	Bits 0 – 15	1 = Alarmed
30065	Output Status: Devices 257-272 (SRAP17 – bit encoded)	Bits 0 – 15	1 = Alarmed
30066	Output Status: Devices 273-288 (SRAP18 – bit encoded)	Bits 0 – 15	1 = Alarmed
30067	Output Status: Devices 289-304 (SRAP19 – bit encoded)	Bits 0 – 15	1 = Alarmed
30068	Output Status: Devices 305-320 (SRAP20 – bit encoded)	Bits 0 – 15	1 = Alarmed
30069	Output Status: Devices 321-336 (SRAP21 – bit encoded)	Bits 0 – 15	1 = Alarmed
30070	Output Status: Devices 337-352 (SRAP22 – bit encoded)	Bits 0 – 15	1 = Alarmed
30071	Output Status: Devices 353-368 (SRAP23 – bit encoded)	Bits 0 – 15	1 = Alarmed
30072	Output Status: Devices 369-384 (SRAP24 – bit encoded)	Bits 0 – 15	1 = Alarmed
30073	Output Status: Devices 385-400 (SRAP25 – bit encoded)	Bits 0 – 15	1 = Alarmed
30074	Output Status: Devices 401-416 (SRAP26 – bit encoded)	Bits 0 – 15	1 = Alarmed
30075	Output Status: Devices 417-432 (SRAP27 – bit encoded)	Bits 0 – 15	1 = Alarmed
30076	Output Status: Devices 433-448 (SRAP28 – bit encoded)	Bits 0 – 15	1 = Alarmed
30077	Output Status: Devices 449-464 (SRAP29 – bit encoded)	Bits 0 – 15	1 = Alarmed
30078	Output Status: Devices 465-480 (SRAP30 – bit encoded)	Bits 0 – 15	1 = Alarmed
30079	Output Status: Devices 481-496 (SRAP31 – bit encoded)	Bits 0 – 15	1 = Alarmed
30080	Output Status: Devices 497-512 (SRAP32 – bit encoded)	Bits 0 – 15	1 = Alarmed
30081	Output Status: Devices 513-528 (SRAP33 – bit encoded)	Bits 0 – 15	1 = Alarmed
30082	Output Status: Devices 529-544 (SRAP34 – bit encoded)	Bits 0 – 15	1 = Alarmed
30083	Output Status: Devices 545-560 (SRAP35 – bit encoded)	Bits 0 – 15	1 = Alarmed
30084	Output Status: Devices 561-576 (SRAP36 – bit encoded)	Bits 0 – 15	1 = Alarmed
30085	Output Status: Devices 577-592 (SRAP37 – bit encoded)	Bits 0 – 15	1 = Alarmed
30086	Output Status: Devices 593-608 (SRAP38 – bit encoded)	Bits 0 – 15	1 = Alarmed
30087	Output Status: Devices 609-624 (SRAP39 – bit encoded)	Bits 0 – 15	1 = Alarmed
30088	Output Status: Devices 625-640 (SRAP40 – bit encoded)	Bits 0 – 15	1 = Alarmed
30089	Output Status: Devices 641-656 (SRAP41 – bit encoded)	Bits 0 – 15	1 = Alarmed
30090	Output Status: Devices 657-672 (SRAP42 – bit encoded)	Bits 0 – 15	1 = Alarmed
30091	Output Status: Devices 673-688 (SRAP43 – bit encoded)	Bits 0 – 15	1 = Alarmed
30092	Output Status: Devices 689-704 (SRAP44 – bit encoded)	Bits 0 – 15	1 = Alarmed
30093	Output Status: Devices 705-720 (SRAP45 – bit encoded)	Bits 0 – 15	1 = Alarmed
30094	Output Status: Devices 721-736 (SRAP46 – bit encoded)	Bits 0 – 15	1 = Alarmed
30095	Output Status: Devices 737-752 (SRAP47 – bit encoded)	Bits 0 – 15	1 = Alarmed
30096	Output Status: Devices 753-768 (SRAP48 – bit encoded)	Bits 0 – 15	1 = Alarmed



30097	Output Status: Devices 769-784 (SRAP49 – bit encoded)	Bits 0 – 15	1 = Alarmed
30098	Output Status: Devices 785-800 (SRAP50 – bit encoded)	Bits 0 – 15	1 = Alarmed
30099	Output Status: Devices 801-816 (SRAP51 – bit encoded)	Bits 0 – 15	1 = Alarmed
30100	Output Status: Devices 817-832 (SRAP52 – bit encoded)	Bits 0 – 15	1 = Alarmed
30101	Output Status: Devices 833-848 (SRAP53 – bit encoded)	Bits 0 – 15	1 = Alarmed
30102	Output Status: Devices 849-864 (SRAP54 – bit encoded)	Bits 0 – 15	1 = Alarmed
30103	Output Status: Devices 865-880 (SRAP55 – bit encoded)	Bits 0 – 15	1 = Alarmed
30104	Output Status: Devices 881-896 (SRAP56 – bit encoded)	Bits 0 – 15	1 = Alarmed
30105	Output Status: Devices 897-912 (SRAP57 – bit encoded)	Bits 0 – 15	1 = Alarmed
30106	Output Status: Devices 913-928 (SRAP58 – bit encoded)	Bits 0 – 15	1 = Alarmed
30107	Output Status: Devices 929-944 (SRAP59 – bit encoded)	Bits 0 – 15	1 = Alarmed
30108	Output Status: Devices 945-960 (SRAP60 – bit encoded)	Bits 0 – 15	1 = Alarmed
30109	Output Status: Devices 961-976 (SRAP61 – bit encoded)	Bits 0 – 15	1 = Alarmed
30110	Output Status: Devices 977-992 (SRAP62 – bit encoded)	Bits 0 – 15	1 = Alarmed
30111	Output Status: Devices 993-1008 (SRAP63 – bit encoded)	Bits 0 – 15	1 = Alarmed
30112	Output Status: Devices 1009-1024 (SRAP64 – bit encoded)	Bits 0 – 15	1 = Alarmed
30113	<i>Reserved for future SCPU registers</i>		
:			
30200			

Note: The fire table registers (shown in shaded registers 30019 – 30048) are only available when DIP switch SW1-8 is on and only on a DCSi licensed for one GRC.

5.3.7 System Input Register

The following registers are reserved for DCSi status information.

Reg#	Addr	Description	Range	Units
39994	N/a	NWL Link Communication Status	0-1	Disabled, Enabled
39995	N/a	DIP Switch Configuration	0-255	
39996	N/a	DCSi Uptime Days counter	0-65535	Days
39997	N/a	DCSi Uptime Hours counter	0-24	Hours
39998	N/a	DCSi Uptime Seconds counter	0-3600	Seconds
39999		DCSi Firmware Version ID	0-999	Version x 100
40000		Spare		

5.3.7.1 Holding Registers

The holding registers can be read and written. Each SCPU is assigned a block of holding register numbers as shown below:

Register # Range	SCPU Address
40001 – 40200	1
40201 – 40400	2



The function of each register is shown in the following table. The register numbers in the table are for SCPU address 1. By offsetting the register addresses in multiples of two hundred, other SCPU's are addressed.

Register #	Description	Valid Range	Units
40001	Opacity Spike Detect Flow Delay	1 - 99	Seconds
40002	Opacity Spike % Over Average	1 - 99	%
40003	Boiler Load Full Scale	0 - 9999	
40004	Alarm Relay Logic	0 - 1	N.O., N.C.
40005	# of Alarms Before Activating Relay	1 - 99	
40006	Program Restart Mode	0 - 1	Continue, Restart
40007	System Clock Hours	0 - 23	Hours (Note 1,2)
40008	System Clock Minutes	0 - 59	Minutes (Note 1,2)
40009	System Clock Seconds	0 - 59	Seconds (Note 1,2)
40010	System Clock Month	1 - 12	Month (Note 1,2)
40011	System Clock Day	1 - 31	Day (Note 1,2)
40012	System Clock Year	0 - 99	Year 0 = 2000 (Note 1,2)
40013	Password (2 characters)	00 - ZZ	ASCII
40014	Active Operating Program	1 - 5	Prog #
40015	Operating/POR Program On/Off	0 - 1	Off, On (Note 4)
40016	Clear Opacity Spike Suspect List	1	Clears list
40017	Auto-Program On/Off Switch	0 - 1	Off, On
40018	Auto-Program by Load - Step 1 Min Load	0 - 9999	
40019	Auto-Program by Load - Step 1 Max Load	0 - 9999	
40020	Auto-Program by Load - Step 1 Program #	0 - 5	None, Prog #
40021	Auto-Program by Load - Step 2 Min Load	0 - 9999	
40022	Auto-Program by Load - Step 2 Max Load	0 - 9999	
40023	Auto-Program by Load - Step 2 Program #	0 - 5	None, Prog #
40024	Auto-Program by Load - Step 3 Min Load	0 - 9999	
40025	Auto-Program by Load - Step 3 Max Load	0 - 9999	
40026	Auto-Program by Load - Step 3 Program #	0 - 5	None, Prog #
40027	Auto-Program by Time - Step 1 Min Time Hours	0 - 23	Hours (Note 3)
40028	Auto-Program by Time - Step 1 Min Time Minutes	0 - 59	Minute (Note 3)
40029	Auto-Program by Time - Step 1 Min Time Seconds	0 - 59	Seconds (Note 3)
40030	Auto-Program by Time - Step 1 Max Time Hours	0 - 23	Hours (Note 3)
40031	Auto-Program by Time - Step 1 Max Time Minutes	0 - 59	Minutes (Note 3)
40032	Auto-Program by Time - Step 1 Max Time Seconds	0 - 59	Seconds (Note 3)
40033	Auto-Program by Time - Step 1 Program #	0 - 5	None, Prog #
40034	Auto-Program by Time - Step 2 Min Time Hours	0 - 23	Hours (Note 3)
40035	Auto-Program by Time - Step 2 Min Time Minutes	0 - 59	Minutes (Note 3)
40036	Auto-Program by Time - Step 2 Min Time Seconds	0 - 59	Seconds (Note 3)
40037	Auto-Program by Time - Step 2 Max Time Hours	0 - 23	Hours (Note 3)
40038	Auto-Program by Time - Step 2 Max Time Minutes	0 - 59	Minutes (Note 3)
40039	Auto-Program by Time - Step 2 Max Time Seconds	0 - 59	Seconds (Note 3)
40040	Auto-Program by Time - Step 2 Program #	0 - 5	None, Prog #
40041	Auto-Program by Time - Step 3 Min Time Hours	0 - 23	Hours (Note 3)
40042	Auto-Program by Time - Step 3 Min Time Minutes	0 - 59	Minutes (Note 3)
40043	Auto-Program by Time - Step 3 Min Time Seconds	0 - 59	Seconds (Note 3)



40044	Auto-Program by Time - Step 3 Max Time Hours	0 - 23	Hours (Note 3)
40045	Auto-Program by Time - Step 3 Max Time Minutes	0 - 59	Minutes (Note 3)
40046	Auto-Program by Time - Step 3 Max Time Seconds	0 - 59	Seconds (Note 3)
40047	Auto-Program by Time - Step 3 Program #	0 - 5	None, Prog #
40048	Auto-Program Default Program	0 - 5	Previous, Prog #
40049	Power Feed X1 Start Slot	0, 1 - 64	Unused, Slot # READ-ONLY
40050	Power Feed X1 Stop Slot	0, 1 - 64	Unused, Slot # READ-ONLY
:	:		
40059	Power Feed X6 Start Slot	0, 1 - 64	Unused, Slot # READ-ONLY
40060	Power Feed X6 Stop Slot	0, 1 - 64	Unused, Slot # READ-ONLY
40061	POR Mode	0 - 2	Off, local, remote
40062	Global Anti-Coincidence Time Reset (Resets all Associated Program ACG Timer to sent value)	0 - 99	Seconds
40063	SCPU Power-up Hours	0 - 23	Hours (Note 2) READ-ONLY
40064	SCPU Power-up Minutes	0 - 59	Minutes (Note 2) READ-ONLY
40065	SCPU Power-up Seconds	0 - 59	Seconds (Note 2) READ-ONLY
40066	SCPU Power-up Month	1 - 12	Month (Note 2) READ-ONLY
40067	SCPU Power-up Day	1 - 31	Day (Note 2) READ-ONLY
40068	SCPU Power-up Year	0 - 99	Year 0 = 2000 (Note 2) READ-ONLY
40069	Anti-Coincidence Time for Program 1	0 - 99	Seconds
40070	Anti-Coincidence Time for Program 2	0 - 99	Seconds
40071	Anti-Coincidence Time for Program 3	0 - 99	Seconds
40072	Anti-Coincidence Time for Program 4	0 - 99	Seconds
40073	Anti-Coincidence Time for Program 5	0 - 99	Seconds
40074	<i>Reserved for future SCPU registers</i>		
:			
40200			

Notes:

1. The clock registers – hours, minutes, seconds, month, day, year – must be written in one transaction using a preset multiple register command. This guarantees that the clock will be updated atomically. All six registers must be read in one transaction also. Attempting to read only part of the time registers will result in zeros being returned.
2. The clear can only be initiated when the DCSi is in the program mode. It should be done prior to downloading an Operating or POR program.
3. Auto-Program by Time Registers – The ‘Min Time’ and ‘Max Time’ registers must be read and written to with all three registers (HH:MM:SS) in the same transaction. For example, to read the step 1 minimum time, your starting register would be 40027 and you would read three registers.
4. Register 40015 – Write a ‘1’ to this register only once to turn the operating/POR program on. Repeatedly writing a ‘1’ to this register will act as program restarts and defeat the operation of the



program, resulting in rappers not firing.

5.3.8 System Holding Register

The following registers are reserved for controlling the DCSi.

Register #	Description	Valid Range	Units
49999 Note 1	DCSi Reset Command Clears all memory locations and reinitializes the UARTs.	1	N/A
49998	DCSi Operating Mode	0 – 3	0 = Operation overview 1 = Rapper firing monitor 2 = Program upload 3 = Program download
49997	DCSi Operation Mode SCPU address	1 – 16	N/A
49996	DCSi ENTER DIAG MODE Command Reserved for NWL use	0-1	N/A

Note 1:

Writing a value = 1 via Function code 06 to register 49999 Clears all memory locations and reinitializes the UARTs.

