FactoryFloor

	OptoControl Command	OptoScript Equivalent (Arguments)
Digital Point	Clear All Latches	`ClearAllLatches(On I/O Unit)`
	Clear Counter	`ClearCounter(On Point)`
	Clear Off-Latch	`ClearOffLatch(On Point)`
	Clear On-Latch	`ClearOnLatch(On Point)`
	Clear Quadrature Counter	`ClearQuadratureCounter(On Point)`
	Generate N Pulses*	`GenerateNPulses(On Time (Seconds), Off Time (Seconds), Number of Pulses, On Point)`
	Get & Clear Counter	`GetClearCounter(From Point)`
	Get & Clear Off-Latch	`GetClearOffLatch(From Point)`
	Get & Clear On-Latch	`GetClearOnLatch(From Point)`
	Get & Clear Quadrature Counter	`GetClearQuadratureCounter(From Point)`
	Get & Restart Off-Pulse Measurement*	`GetRestartOffPulseMeasurement(From Point)`
	Get & Restart Off-Time Totalizer*	`GetRestartOffTimeTotalizer(From Point)`
	Get & Restart On-Pulse Measurement*	`GetRestartOnPulseMeasurement(From Point)`
	Get & Restart On-Time Totalizer*	`GetRestartOnTimeTotalizer(From Point)`
	Get & Restart Period*	`GetRestartPeriod(From Point)`
	Get Counter	`GetCounter(From Point)`
	Get Frequency	`GetFrequency(From Point)`
	Get Off-Latch	*See* Off-Latch Set?
	Get Off-Pulse Measurement*	`GetOffPulseMeasurement(From Point)`
	Get Off-Pulse Measurement Complete Status*	`GetOffPulseMeasurementCompleteStatus(From Point)`
	Get Off-Time Totalizer*	`GetOffTimeTotalizer(From Point)`
	Get On-Latch	*See* On-Latch Set?
	Get On-Pulse Measurement*	`GetOnPulseMeasurement(From Point)`
	Get On-Pulse Measurement Complete Status*	`GetOnPulseMeasurementCompleteStatus(From Point)`
	Get On-Time Totalizer*	`GetOnTimeTotalizer(From Point)`
	Get Period*	`GetPeriod(From Point)`
	Get Period Measurement Complete Status*	`GetPeriodMeasurementCompleteStatus(From Point)`
	Get Quadrature Counter	`GetQuadratureCounter(On Point)`
	Off?	`IsOff(Point)`
	Off-Latch Set?	`IsOffLatchSet(On Point)`
	On?	`IsOn(Point)`
	On-Latch Set?	`IsOnLatchSet(On Point)`
	Set TPO Percent*	`SetTpoPercent(To Percent, On Point)`
	Set TPO Period*	`SetTpoPeriod(To Seconds, On Point)`
	Start Continuous Square Wave*	`StartContinuousSquareWave(On Time (Seconds), Off Time (Seconds), On Point)`
	Start Counter	`StartCounter(On Point)`
	Start Off-Pulse*	`StartOffPulse(Off Time (Seconds), On Point)`
	Start On-Pulse*	`StartOnPulse(On Time (Seconds), On Point)`
	Start Quadrature Counter	`StartQuadratureCounter(On Point)`
	Stop Counter	`StopCounter(On Point)`
	Stop Quadrature Counter	`StopQuadratureCounter(On Point)`
	Turn Off	`TurnOff(Output)`
	Turn On	`TurnOn(Output)`

**Not available on SNAP Ethernet-based I/O units*

	OptoControl Command	OptoScript Equivalent (Arguments)
PID	Clamp PID Output*	`ClampPidOutput(High Clamp, Low Clamp, On PID Loop)`
	Clamp PID Setpoint*	`ClampPidSetpoint(High Clamp, Low Clamp, On PID Loop)`
	Disable PID Output*	`DisablePidOutput(Of PID Loop)`
	Disable PID Output Tracking in Manual Mode*	`DisablePidOutputTrackingInManualMode(On PID Loop)`
	Disable PID Setpoint Tracking in Manual Mode*	`DisablePidSetpointTrackingInManualMode(On PID Loop)`
	Enable PID Output*	`EnablePidOutput(Of PID Loop)`
	Enable PID Output Tracking in Manual Mode*	`EnablePidOutputTrackingInManualMode(On PID Loop)`
	Enable PID Setpoint Tracking in Manual Mode*	`EnablePidSetpointTrackingInManualMode(On PID Loop)`
	Get PID Control Word*	`GetPidControlWord(From PID Loop)`
	Get PID D Term*	`GetPidDTerm(From PID Loop)`
	Get PID I Term*	`GetPidITerm(From PID Loop)`
	Get PID Input*	`GetPidInput(From PID Loop)`
	Get PID Mode*	`GetPidMode(From PID Loop)`
	Get PID Output*	`GetPidOutput(From PID Loop)`
	Get PID Output Rate of Change*	`GetPidOutputRateOfChange(From PID Loop)`
	Get PID P Term*	`GetPidPTerm(From PID Loop)`
	Get PID Scan Rate*	`GetPidScanRate(From PID Loop)`
	Get PID Setpoint*	`GetPidSetpoint(From PID Loop)`
	Set PID Control Word*	`SetPidControlWord(On Mask, Off Mask, For PID Loop)`
	Set PID D Term*	`SetPidDTerm(To, On PID Loop)`
	Set PID I Term*	`SetPidITerm(To, On PID Loop)`
	Set PID Input*	`SetPidInput(To, On PID Loop)`
	Set PID Mode to Auto*	`SetPidModeToAuto(On PID Loop)`
	Set PID Mode to Manual*	`SetPidModeToManual(On PID Loop)`
	Set PID Output Rate of Change*	`SetPidOutputRateOfChange(To, On PID Loop)`
	Set PID P Term*	`SetPidPTerm(To, On PID Loop)`
	Set PID Scan Rate*	`SetPidScanRate(To, On PID Loop)`
	Set PID Setpoint*	`SetPidSetpoint(To, On PID Loop)`

**Not available on SNAP Ethernet brains*

	OptoControl Command	OptoScript Equivalent (Arguments)
Simulation	Communication to All I/O Points Enabled?	`IsCommToAllIoPointsEnabled()`
	Communication To All I/O Units Enabled?	`IsCommToAllIoUnitsEnabled()`
	Disable Communication to All I/O Points	`DisableCommuncationToAllIoPoints()`
	Disable Communication to All I/O Units	`DisableCommunicationToAllIoUnits()`
	Disable Communication to Analog Point	`DisableCommunicationToAnalogPoint(`*`Analog Point`*`)`
	Disable Communication to Digital Point	`DisableCommunicationToDigitalPoint(`*`Digital Point`*`)`
	Disable Communication to Event/Reaction	`DisableCommunicationToEventReaction(`*`Event/Reaction`*`)`
	Disable Communication to I/O Unit	`DisableCommunicationToIoUnit(`*`I/O Unit`*`)`
	Disable Communication to PID Loop	`DisableCommunicationToPidLoop(`*`PID Loop`*`)`
	Disable Event/Reaction Group	`DisableEventReactionGroup(`*`E/R Group`*`)`
	Enable Communication to All I/O Points	`EnableCommunicationToAllIoPoints()`
	Enable Communication to All I/O Units	`EnableCommunicationToAllIoUnits()`
	Enable Communication to Analog Point	`EnableCommunicationToAnalogPoint(`*`Analog Point`*`)`
	Enable Communication to Digital Point	`EnableCommunicationToDigitalPoint(`*`Digital Point`*`)`
	Enable Communication to Event/Reaction	`EnableCommunicationToEventReaction(`*`Event/Reaction`*`)`
	Enable Communication to I/O Unit	`EnableCommunicationToIoUnit(`*`I/O Unit`*`)`
	Enable Communication to PID Loop	`EnableCommunicationToPidLoop(`*`PID Loop`*`)`
	Enable Event/Reaction Group	`EnableEventReactionGroup(`*`E/R Group`*`)`
	Event/Reaction Communication Enabled?	`IsEventReactionCommEnabled(`*`Event/Reaction`*`)`
	Event/Reaction Group Communication Enabled?	`IsEventReactionGroupEnabled(`*`E/R Group`*`)`
	I/O Point Communication Enabled?	`IsIoPointCommEnabled(`*`I/O Point`*`)`
	I/O Unit Communication Enabled?	`IsIoUnitCommEnabled(`*`I/O Unit`*`)`
	IVAL Set Analog from Table	`IvalSetAnalogFromTable(`*`Start at Index, Of Table, On I/O Unit`*`)`
	IVAL Set Analog Point	`IvalSetAnalogPoint(`*`To, On Point`*`)`
	IVAL Set Counter	`IvalSetCounter(`*`To, On Point`*`)`
	IVAL Set Digital Binary	`IvalSetDigitalBinary(`*`On Mask, Off Mask, On I/O Unit`*`)`
	IVAL Set Frequency	`IvalSetFrequency(`*`To, On Point`*`)`
	IVAL Set Off-Latch	`IvalSetOffLatch(`*`To, On Point`*`)`
	IVAL Set Off-Pulse	`IvalSetOffPulse(`*`To, On Point`*`)`
	IVAL Set Off-Totalizer	`IvalSetOffTotalizer(`*`To, On Point`*`)`
	IVAL Set On-Latch	`IvalSetOnLatch(`*`To, On Point`*`)`
	IVAL Set On-Pulse	`IvalSetOnPulse(`*`To, On Point`*`)`
	IVAL Set On-Totalizer	`IvalSetOnTotalizer(`*`To, On Point`*`)`
	IVAL Set Period	`IvalSetPeriod(`*`To, On Point`*`)`
	IVAL Set PID Control Word	`IvalSetPidControlWord(`*`On Mask, Off Mask, For PID Loop`*`)`
	IVAL Set PID Process Term	`IvalSetPidProcessTerm(`*`To, On PID Loop`*`)`
	IVAL Set Quadrature Counter	`IvalSetQuadratureCounter(`*`To, On Point`*`)`
	IVAL Set TPO Percent	`IvalSetTpoPercent(`*`To, On Point`*`)`
	IVAL Set TPO Period	`IvalSetTpoPeriod(`*`To, On Point`*`)`
	IVAL Turn Off	`IvalTurnOff(`*`Point`*`)`
	IVAL Turn On	`IvalTurnOn(`*`Point`*`)`
	PID Loop Communication Enabled?	`IsPidLoopCommEnabled(`*`PID Loop`*`)`

	OptoControl Command	OptoScript Equivalent (Arguments)
Analog Point	Calculate & Set Analog Gain	CalcSetAnalogGain(*On Point*)
	Calculate & Set Analog Offset	CalcSetAnalogOffset(*On Point*)
	Get & Clear Analog Filtered Value*	GetClearAnalogFilteredValue(*From*)
	Get & Clear Analog Maximum Value	GetClearAnalogMaxValue(*From*)
	Get & Clear Analog Minimum Value	GetClearAnalogMinValue(*From*)
	Get & Clear Analog Totalizer Value*	GetClearAnalogTotalizerValue(*From*)
	Get Analog Filtered Value*	GetAnalogFilteredValue(*From*)
	Get Analog Lower Clamp	GetAnalogLowerClamp(*From*)
	Get Analog Maximum Value	GetAnalogMaxValue(*From*)
	Get Analog Minimum Value	GetAnalogMinValue(*From*)
	Get Analog Square Root Filtered Value*	GetAnalogSquareRootFilteredValue(*From*)
	Get Analog Square Root Value*	GetAnalogSquareRootValue(*From*)
	Get Analog Totalizer Value*	GetAnalogTotalizerValue(*From*)
	Get Analog Upper Clamp	GetAnalogUpperClamp(*From*)
	Ramp Analog Output*	RampAnalogOutput(*Ramp Endpoint, Units/Sec, Point to Ramp*)
	Set Analog Filter Weight*	SetAnalogFilterWeight(*To, On Point*)
	Set Analog Gain	SetAnalogGain(*To, On Point*)
	Set Analog Offset	SetAnalogOffset(*To, On Point*)
	Set Analog Totalizer Rate*	SetAnalogTotalizerRate(*To Seconds, On Point*)
	Set Analog TPO Period	SetAnalogTpoPeriod(*To, On Point*)

**Not available on SNAP Ethernet-based I/O units*

Chart	Call Chart	CallChart(*Chart*)
	Calling Chart Running?	IsCallingChartRunning()
	Calling Chart Stopped?	IsCallingChartStopped()
	Calling Chart Suspended?	IsCallingChartSuspended()
	Chart Running?	IsChartRunning(*Chart*)
	Chart Stopped?	IsChartStopped(*Chart*)
	Chart Suspended?	IsChartSuspended(*Chart*)
	Continue Calling Chart	ContinueCallingChart()
	Continue Chart	ContinueChart(*Chart*)
	Get Chart Status	GetChartStatus(*Chart*)
	Get Priority	GetPriority()
	Get Priority of Host Task	GetPriorityOfHostTask(*On Port*)
	Host Task Received A Message?	HasHostTaskReceivedMessage(*On Port*)
	Set Priority	SetPriority()
	Set Priority Of Host Task	SetPriorityOfHostTask(*On Port*)
	Start Chart	StartChart(*Chart*)
	Start Default Host Task	StartDefaultHostTask()
	Start Host Task (ASCII)	StartHostTaskAscii(*On Port*)
	Start Host Task (Binary)	StartHostTaskBinary(*On Port*)
	Stop Chart	StopChart(*Chart*)
	Stop Chart on Error	StopChartOnError()
	Stop Host Task	StopHostTask(*On Port*)
	Suspend Chart	SuspendChart(*Chart*)
	Suspend Chart on Error	SuspendChartOnError()
	Suspend Default Host Task	SuspendDefaultHostTask()

	OptoControl Command	OptoScript Equivalent (Arguments)
I/O Unit	Configure I/O Unit	ConfigureIoUnit(*I/O Unit*)
	Get & Clear Digital I/O Unit Latches	GetClearDigitalIoUnitLatches(*From, State, On-Latch, Off-Latch, Clear Flag*)
	Get & Clear Digital-64 I/O Unit Latches	GetClearDigital64IoUnitLatches(*From, State, On-Latch, Off-Latch, Clear Flag*)
	Get & Clear Simple-64 I/O Unit Latches	GetClearSimple64IoUnitLatches(*From, State, On-Latch, Off-Latch, Clear Flag*)
	Get Digital I/O Unit as Binary Value	GetDigitalIoUnitAsBinaryValue(*I/O Unit*)
	Get Digital-64 I/O Unit as Binary Value	GetDigital64IoUnitAsBinaryValue(*I/O Unit*)
	Get Digital I/O Unit Latches	GetDigitalIoUnitLatches(*From, State, On-Latch, Off-Latch*)
	Get Digital-64 I/O Unit Latches	GetDigital64IoUnitLatches(*From, State, On-Latch, Off-Latch*)
	Get Mixed I/O Unit as Binary Value	GetMixedIoUnitAsBinaryValue(*I/O Unit*)
	Get Simple-64 I/O Unit as Binary Value	GetSimple64IoUnitAsBinaryValue(*I/O Unit*)
	Get Simple-64 I/O Unit Latches	GetSimple64IoUnitLatches(*From, State, On-Latch, Off-Latch*)
	I/O Unit Ready?	IsIoUnitReady(*I/O Unit*)
	Move Analog I/O Unit to Table	MoveAnalogIoUnitToTable(*I/O Unit, To Index, Of Table*)
	Move Digital I/O Unit to Table	MoveDigitalIoUnitToTable(*I/O Unit, Starting Index, Of Table*)
	Move Digital I/O Unit to Table Element	(No exact equivalent. See the OptoControl Command Reference for an alternative method.)
	Move Mixed I/O Unit to Table	MoveMixedIoUnitToTable(*I/O Unit, Starting Index, Of Table*)
	Move Simple-64 I/O Unit to Table	MoveSimple64IoUnitToTable(*I/O Unit, Starting Index, Of Table*)
	Move Table Element to Digital I/O Unit	MoveTableElementToDigitalIoUnit(*From Table, Of Table, Move To*)
	Move Table to Analog I/O Unit	MoveTableToAnalogIoUnit(*Start at Index, Of Table, Move to*)
	Move Table to Digital I/O Unit	MoveTableToDigitalIoUnit(*Start at Index, Of Table, Move to*)
	Move Table to Mixed I/O Unit	MoveTableToMixedIoUnit(*Start at Index, Of Table, Move to*)
	Move Table to Simple-64 I/O Unit	MoveTableToSimple64IoUnit(*Start at Index, Of Table, Move to*)
	Set Digital I/O Unit from MOMO Masks	SetDigitalIoUnitFromMomo(*Must-On Mask, Must-Off Mask, Digital I/O Unit*)
	Set Digital-64 I/O Unit from MOMO Masks	SetDigital64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Digital-64 I/O Unit*)
	Set I/O Unit Configured Flag	SetIoUnitConfiguredFlag(*For I/O Unit*)
	Set Mixed I/O Unit from MOMO Masks	SetMixedIoUnitFromMomo(*Must-On Mask, Must-Off Mask, Mixed I/O Unit*)
	Set Number of Retries to All I/O Units	SetNumberOfRetriesToAllIoUnits(*To*)
	Set Simple-64 I/O Unit from MOMO Masks	SetSimple64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Simple-64 I/O Unit*)
	Write I/O Unit Configuration to EEPROM	WriteIoUnitConfigToEeprom(*On I/O Unit*)

1342-041206—OptoControl 4.1 Commands Quick Reference
Copyright © 1999–2004 Opto 22

	OptoControl Command	OptoScript Equivalent (Arguments)
Miscellaneous	Comment (Block)	`/* block comment */`
	Comment (Single Line)	`// single line comment`
	Continue Timer	`ContinueTimer(Timer)`
	Delay (mSec)	`DelayMsec(Milliseconds)`
	Delay (Sec)	`DelaySec(Seconds)`
	Down Timer Expired?	`HasDownTimerExpired(Down Timer)`
	Float Valid?	`IsFloatValid(Float)`
	Generate Reverse CRC-16 on Table (32 bit)	`GenerateReverseCrc16OnTable32(Start Value, Table, Starting Element, Number of Elements)`
	Get Length of Table	`GetLengthOfTable(Table)`
	Move	`x = y;`
	Move from Table Element	`x = nt[0];`
	Move Table Element to Table	`nt1[0] = nt2[5];`
	Move Table to Table	`MoveTableToTable(From Table, From Index, To Table, To Index, Length)`
	Move to Table Element	`nt[0] = x;`
	Pause Timer	`PauseTimer(Timer)`
	Set Down Timer Preset Value	`SetDownTimerPreset(Target Value, Down Timer)`
	Set Up Timer Target Value	`SetUpTimerTarget(Target Value, Up Timer)`
	Shift Table Elements	`ShiftTableElements(Shift Count, Table)`
	Start Timer	`StartTimer(Timer)`
	Stop Timer	`StopTimer(Timer)`
	Timer Expired?	`HasTimerExpired(Timer)`
	Up Timer Target Time Reached?	`HasUpTimerReachedTargetTime(Up Timer)`
Event/Reaction	Clear All Event Latches	`ClearAllEventLatches(On I/O Unit)`
	Clear Event Latch	`ClearEventLatch(On Event/Reaction)`
	Clear I/O Unit Interrupt	`ClearIoUnitInterrupt(On I/O Unit)`
	Disable Interrupt On Event	`DisableInterruptOnEvent(Event/Reaction)`
	Disable Scanning For All Events	`DisableScanningForAllEvents(On I/O Unit)`
	Disable Scanning For Event	`DisableScanningForEvent(Event/Reaction)`
	Disable Scanning of Event/Reaction Group	`DisableScanningOfEventReactionGroup(E/R Group)`
	Enable Interrupt on Event	`EnableInterruptOnEvent(Event/Reaction)`
	Enable Scanning For All Events	`EnableScanningForAllEvents(On I/O Unit)`
	Enable Scanning For Event	`EnableScanningForEvent(Event/Reaction)`
	Enable Scanning of Event/Reaction Group	`EnableScanningOfEventReactionGroup(E/R Group)`
	Event Occurred?	`HasEventOccurred(Event/Reaction)`
	Event Occurring?	`IsEventOccurring(Event/Reaction)`
	Event Scanning Disabled?	`IsEventScanningDisabled(Event/Reaction)`
	Event Scanning Enabled?	`IsEventScanningEnabled(Event/Reaction)`
	Generating Interrupt?	`IsGeneratingInterrupt(I/O Unit)`
	Get & Clear Event Latches	`GetClearEventLatches(E/R Group)`
	Get Event Latches	`GetEventLatches(E/R Group)`
	Interrupt Disabled For Event?	`IsInterruptDisabledForEvent(Event/Reaction)`
	Interrupt Enabled For Event?	`IsInterruptEnabledForEvent(Event/Reaction)`
	Read Event/Reaction Hold Buffer	`ReadEventReactionHoldBuffer(Event/Reaction)`

	OptoControl Command	OptoScript Equivalent (Arguments)
Controller	Add User Error to Queue	`AddUserErrorToQueue(Error Number)`
	Add User I/O Unit Error to Queue	`AddUserIoUnitErrorToQueue(Error Number, I/O Unit)`
	Calculate & Store Srategy CRC	`CalcStoreStrategyCRC()`
	Calculate Strategy CRC	`CalcStrategyCrc()`
	Caused a Chart Error?	`HasChartCausedError(Chart)`
	Caused an I/O Unit Error?	`HasIoUnitCausedError(I/O Unit)`
	Clear All Errors	`ClearAllErrors()`
	Clear PC Byte Swap Mode (ISA only)	`ClearPcByteSwapMode()`
	Disable I/O Unit Causing Current Error	`DisableIoUnitCausingCurrentError()`
	Enable I/O Unit Causing Current Error	`EnableIoUnitCausingCurrentError()`
	Error?	`IsErrorPresent()`
	Error on I/O Unit?	`IsErrorOnIoUnit()`
	Get Address of I/O Unit Causing Current Error	`GetAddressOfIoUnitCausingCurrentError()`
	Get Controller Address	`GetControllerAddress()`
	Get Controller Type	`GetControllerType()`
	Get Default Host Port	`GetDefaultHostPort()`
	Get Error Code of Current Error	`GetErrorCodeOfCurrentError()`
	Get Error Count	`GetErrorCount()`
	Get Firmware Version	`GetFirmwareVersion(Put in)`
	Get ID of Block Causing Current Error	`GetIdOfBlockCausingCurrentError()`
	Get Name of Chart Causing Current Error	`GetNameOfChartCausingCurrentError(Put in)`
	Get Name of I/O Unit Causing Current Error	`GetNameOfIoUnitCausingCurrentError(Put in)`
	Get Port of I/O Unit Causing Current Error	`GetPortOfIoUnitCausingCurrentError()`
	Get RTU/M4IO Temperature	`GetRtuM4IoTemperature()`
	Get RTU/M4IO Voltage	`GetRtuM4IoVoltage()`
	Low RAM Backup Battery?	`IsRamBackupBatteryLow()`
	Read Byte from PC Memory (ISA only)	`ReadByteFromPcMemory(From Address)`
	Read Byte from PC Port (ISA only)	`ReadByteFromPcPort(From Address)`
	Read Word from PC Memory (ISA only)	`ReadWordFromPcMemory(From Address)`
	Read Word from PC Port (ISA only)	`ReadWordFromPcPort(From Address)`
	Remove Current Error and Point to Next Error	`RemoveCurrentError()`
	Reset Controller	`ResetController()`
	Retrieve Strategy CRC	`RetrieveStrategyCrc()`
	Set PC Byte Swap Mode (ISA only)	`SetPcByteSwapMode()`
	Write Byte to PC Memory (ISA only)	`WriteByteToPcMemory(Byte, To Address)`
	Write Byte to PC Port (ISA only)	`WriteByteToPcPort(Byte, To Address)`
	Write Word to PC Memory (ISA only)	`WriteWordToPcMemory(Word, To Address)`
	Write Word to PC Port (ISA only)	`WriteWordToPcPort(Word, To Address)`

1342-041206—OptoControl 4.1 Commands Quick Reference
Copyright © 1999–2004 Opto 22

	OptoControl Command	OptoScript Equivalent (Arguments)
Time/Date	Copy Date to String (DD/MM/YY)	DateToStringDDMMYY(*String*)
	Copy Date to String (MM/DD/YY)	DateToStringMMDDYY(*String*)
	Copy Time to String	TimeToString(*String*)
	Get Day	GetDay()
	Get Day of Week	GetDayOfWeek()
	Get Hours	GetHours()
	Get Julian Day	GetJulianDay()
	Get Minutes	GetMinutes()
	Get Month	GetMonth()
	Get Seconds	GetSeconds()
	Get Seconds Since Midnight	GetSecondsSinceMidnight()
	Get System Time	GetSystemTime()
	Get Year	GetYear()
	Set Date	SetDate(*To*)
	Set Day	SetDay(*To*)
	Set Day of Week	SetDayOfWeek(*To*)
	Set Hours	SetHours(*To*)
	Set Minutes	SetMinutes(*To*)
	Set Month	SetMonth(*To*)
	Set Seconds	SetSeconds(*To*)
	Set Time	SetTime(*To*)
	Set Year	SetYear(*To*)
Communication—I/O	Convert Mistic I/O Hex to Float	MisticIoHexToFloat(*Convert*)
	Convert Number to Mistic I/O Hex	NumberToMisticIoHex(*Convert, Put Result in*)
	Read Numeric Table from I/O Memory Map	ReadNumTableFromIoMemMap(*Length, Start Index, I/O Unit, Mem address, To*)
	Read Numeric Variable from I/O Memory Map	ReadNumVarFromIoMemMap(*I/O Unit, Mem address, To*)
	Read String Table from I/O Memory Map	ReadStrTableFromIoMemMap(*Length, Start Index, I/O Unit, Mem address, To*)
	Read String Variable from I/O Memory Map	ReadStrVarFromIoMemMap(*Length, I/O Unit, Mem address, To*)
	Transmit/Receive Mistic I/O Hex String with Checksum	TransReceMisticIoHexStringWithChecksum(*Hex String, On Port, Put Result in*)
	Transmit/Receive Mistic I/O Hex String with CRC	TransReceMisticIoHexStringWithCrc(*Hex String, On Port, Put Result in*)
	Transmit/Receive OPTOMUX String	TransReceOptomuxString(*String, On Port, Put Result in*)
	Write Numeric Table to I/O Memory Map	WriteNumTableToIoMemMap(*Length, Start Index, I/O Unit, Mem address, Table*)
	Write Numeric Variable to I/O Memory Map	WriteNumVarToIoMemMap(*I/O Unit, Mem address, Variable*)
	Write String Table to I/O Memory Map	WriteStrTableToIoMemMap(*Length, Start Index, I/O Unit, Mem address, Table*)
	Write String Variable to I/O Memory Map	WriteStrVarToIoMemMap(*I/O Unit, Mem address, Variable*)

	OptoControl Command	OptoScript Equivalent (Arguments)
Communication—Network	Accept Session on TCP Port	AcceptSessionOnTcpPort(*TCP Port*)
	ARCNET Connected?	IsArcnetConnected()
	ARCNET Message Address Equal To?	IsArcnetMsgAddressEqual(*Address*)
	ARCNET Node Present?	IsArcnetNodePresent(*Address*)
	Close Ethernet Session	CloseEthernetSession(*Session, On Port*)
	Ethernet Session Open?	IsEnetSessionOpen(*Session*)
	Get ARCNET Destination Address on Port	GetArcnetDestAddressOnPort(*On Port*)
	Get ARCNET Host Destination Address	GetArcnetHostDestAddress()
	Get ARCNET Peer Destination Address	GetArcnetPeerDestAddress()
	Get Ethernet Session Name	GetEthernetSessionName(*Session, Put in*)
	Get Number of Characters Waiting on Ethernet Session	GetNumCharsWaitingOnEnetSession(*On Session*)
	Open Ethernet Session	OpenEthernetSession(*Session Name, On Port*)
	Receive N Characters via ARCNET	ReceiveNCharsViaArcnet(*Put in, Num. Characters, From Port*)
	Receive N Characters via Ethernet	ReceiveNCharsViaEthernet(*Put in, Num. Characters, From Session*)
	Receive String via ARCNET	ReceiveStringViaArcnet(*Put in, From Port*)
	Receive String via Ethernet	ReceiveStringViaEthernet(*Put in, From Session*)
	Receive Table via ARCNET	ReceiveTableViaArcnet(*Start at Index, Of Table, From Port*)
	Receive Table via Ethernet	ReceiveTableViaEthernet(*Start at Index, Of Table, From Session*)
	Set ARCNET Destination Address on Port	SetArcnetDestAddressOnPort(*To Address, On Port*)
	Set ARCNET Host Destination Address	SetArcnetHostDestAddress(*To*)
	Set ARCNET Mode Raw	SetArcnetModeRaw()
	Set ARCNET Mode Standard	SetArcnetModeStandard()
	Set ARCNET Peer Destination Address	SetArcnetPeerDestAddress(*To*)
	Transmit String via ARCNET	TransStringViaArcnet(*String, On Port*)
	Transmit String via Ethernet	TransStringViaEthernet(*String, Via Session, On Port*)
	Transmit Table via ARCNET	TransTableViaArcnet(*Start at Index, Of Table, On Port*)
	Transmit Table via Ethernet	TransTableViaEthernet(*Start at Index, Of Table, Via Session, On Port*)
	Transmit/Receive String via ARCNET	TransReceStringViaArcnet(*String, On Port, Put Result in*)
	Transmit/Receive String via Ethernet	TransReceStringViaEthernet(*String, Via Session, On Port, Put Result in*)

1342-041206—OptoControl 4.1 Commands Quick Reference
Copyright © 1999–2004 Opto 22

	OptoControl Command	OptoScript Equivalent (Arguments)
Mathematical	Absolute Value	`AbsoluteValue(Of)`
	Add	`x + y`
	Arccosine	`Arccosine(Of)`
	Arcsine	`Arcsine(Of)`
	Arctangent	`Arctangent(Of)`
	Clamp Float Table Element	`ClampFloatTableElement(High Limit, Low Limit, Element Index, Of Float Table)`
	Clamp Float Variable	`ClampFloatVariable(High Limit, Low Limit, Float Variable)`
	Clamp Integer 32 Table Element	`ClampInt32TableElement(High Limit, Low Limit, Element Index, Of Integer 32 Table)`
	Clamp Integer 32 Variable	`ClampInt32Variable(High Limit, Low Limit, Integer 32 Variable)`
	Complement	`-x`
	Cosine	`Cosine(Of)`
	Decrement Variable	`DecrementVariable(Variable)`
	Divide	`x / y`
	Generate Random Number	`GenerateRandomNumber()`
	Hyperbolic Cosine	`HyperbolicCosine(Of)`
	Hyperbolic Sine	`HyperbolicSine(Of)`
	Hyperbolic Tangent	`HyperbolicTangent(Of)`
	Increment Variable	`IncrementVariable(Variable)`
	Maximum	`Max(Compare, With)`
	Minimum	`Min(Compare, With)`
	Modulo	`x % y`
	Multiply	`x * y`
	Natural Log	`NaturalLog(Of)`
	Raise e to Power	`RaiseEToPower(Exponent)`
	Raise to Power	`Power(Raise, To the)`
	Round	`Round(Value)`
	Seed Random Number	`SeedRandomNumber()`
	Sine	`Sine(Of)`
	Square Root	`SquareRoot(Of)`
	Subtract	`x - y`
	Tangent	`Tangent(Of)`
	Truncate	`Truncate(Value)`

	OptoControl Command	OptoScript Equivalent (Arguments)
String	Append Character to String	`s1 += 'a';`
	Append String to String	`s1 += s2;`
	Convert Float to String	`FloatToString(Convert, Length, Decimals, Put Result in)`
	Convert Hex String to Number	`HexStringToNumber(Convert)`
	Convert IEEE Hex String to Number	`IEEEHexStringToNumber(Convert)`
	Convert Number to Formatted Hex String	`NumberToFormattedHexString(Convert, Length, Put Result in)`
	Convert Number to Hex String	`NumberToHexString(Convert, Put Result in)`
	Convert Number to String	`NumberToString(Convert, Put Result in)`
	Convert Number to String Field	`NumberToStringField(Convert, Length, Put Result in)`
	Convert String to Float	`StringToFloat(Convert)`
	Convert String to Integer 32	`StringToInt32(Convert)`
	Convert String to Integer 64	`StringToInt64(Convert)`
	Convert String to Lower Case	`StringToLowerCase(Convert)`
	Convert String to Upper Case	`StringToUpperCase(Convert)`
	Find Character in String	`FindCharacterInString(Find, Start at Index, Of String)`
	Find Substring in String	`FindSubstringInString(Find, Start at Index, Of String)`
	Generate Checksum on String	`GenerateChecksumOnString(Start Value, On String)`
	Generate Forward CCITT on String	`GenerateForwardCcittOnString(Start Value, On String)`
	Generate Forward CRC-16 on String	`GenerateForwardCrc16OnString(Start Value, On String)`
	Generate Reverse CCITT on String	`GenerateReverseCcittOnString(Start Value, On String)`
	Generate Reverse CRC-16 on String	`GenerateReverseCrc16OnString(Start Value, On String)`
	Get Nth Character	`GetNthCharacter(From String, Index)`
	Get String Length	`GetStringLength(Of String)`
	Get Substring	`GetSubstring(From String, Start at Index, Num. Characters, Put Result in)`
	Move from String Table	`s = st[0];`
	Move String	`s1 = s2;`
	Move to String Table	`st[0] = s;`
	Set Nth Character	`SetNthCharacter(To, In String, At Index)`
	String Equal?	`s1 == s2`
	String Equal to String Table Element?	`s == st[0]`
	Test Equal Strings	*See* String Equal?
	Verify Checksum on String	`VerifyChecksumOnString(Start Value, On String)`
	Verify Forward CCITT on String	`VerifyForwardCcittOnString(Start Value, On String)`
	Verify Forward CRC-16 on String	`VerifyForwardCrc16OnString(Start Value, On String)`
	Verify Reverse CCITT on String	`VerifyReverseCcittOnString(Start Value, On String)`
	Verify Reverse CRC-16 on String	`VerifyReverseCrc16OnString(Start Value, On String)`

Pointers	Clear Pointer	`pn1 = null;`
	Clear Pointer Table Element	`pt[0] = null;`
	Move from Pointer Table Element	`pn = pt[0];`
	Move to Pointer	`pn = &n;`
	Move to Pointer Table	`pt[0] = &n;`
	Pointer Equal to Null?	`pn == null`
	Pointer Table Element Equal to Null?	`pt[0] == null`

1342-041206—OptoControl 4.1 Commands Quick Reference
Copyright © 1999–2004 Opto 22

Logical

OptoControl Command	OptoScript Equivalent (Arguments)
AND	`x and y`
AND?	*See* AND
Bit AND	`x bitand y`
Bit AND?	*See* Bit AND
Bit Clear	`BitClear(Item, Bit to Clear)`
Bit NOT	`bitnot x`
Bit NOT?	*See* Bit NOT
Bit Off?	`IsBitOff(In, Bit)`
Bit On?	`IsBitOn(In, Bit)`
Bit OR	`x bitor y`
Bit OR?	*See* Bit OR
Bit Rotate	`BitRotate(Item, Count)`
Bit Set	`BitSet(Item, Bit to Set)`
Bit Shift	`x << nBitsToShift`
Bit Test	`BitTest(Item, Bit to Test)`
Bit XOR	`x bitxor y`
Bit XOR?	*See* Bit XOR
Equal?	`x == y`
Equal to Table Element?	`n == nt[0]`
Get High Bits of Integer 64	`GetHighBitsOfInt64(High Bits From)`
Get Low Bits of Integer 64	`GetLowBitsOfInt64(Integer 64)`
Greater?	`x > y`
Greater Than Table Element?	`x > nt[0]`
Greater Than or Equal?	`x >= y`
Greater Than or Equal to Table Element?	`x >= nt[0]`
Less?	`x < y`
Less Than Table Element?	`x < nt[0]`
Less Than or Equal?	`x <= y`
Less Than or Equal to Table Element?	`x <= nt[0]`
Make Integer 64	`MakeInt64(High Integer, Low Integer)`
Move 32 Bits	`Move32Bits(From, To)`
NOT	`not x`
NOT?	`not x`
Not Equal?	`x <> y`
Not Equal to Table Element?	`n <> nt[0]`
OR	`x or y`
OR?	*See* OR
Set Variable False	`SetVariableFalse(Variable)`
Set Variable True	`SetVariableTrue(Variable)`
Table Element Bit Clear	`TableElementBitClear(Element Index, Of Integer Table, Bit to Clear)`
Table Element Bit Set	`TableElementBitSet(Element Index, Of Integer Table, Bit to Set)`
Table Element Bit Test	`TableElementBitTest(Element Index, Of Integer Table, Bit to Test)`
Test Equal	*See* Equal?
Test Greater	*See* Greater?
Test Greater or Equal	*See* Greater Than or Equal?
Test Less	*See* Less?
Test Less or Equal	*See* Less Than or Equal?
Test Not Equal	*See* Not Equal?

OptoControl Command	OptoScript Equivalent (Arguments)
Test Within Limits	*See* Within Limits?
Variable False?	`IsVariableFalse(Variable)`
Variable True?	`IsVariableTrue(Variable)`
Within Limits?	`IsWithinLimits(Value, Low Limit, High Limit)`
XOR	`x xor y`
XOR?	*See* XOR

Communication—Serial

Characters Waiting at Serial Port?	`AreCharsWaitingAtSerialPort(Port)`
Clear Receive Buffer	`ClearReceiveBuffer()`
Configure Port	`ConfigurePort(Configuration)`
Configure Port Timeout Delay	`ConfigurePortTimeoutDelay(Delay (Seconds), On Port)`
CTS Off?	`IsCtsOff(On Port)`
CTS On?	`IsCtsOn(On Port)`
Get Active Interrupt Mask	`GetActiveInterruptMask()`
Get Number of Characters Waiting on Serial or ARCNET Port	`GetNumCharsWaitingOnPort(On Port)`
Interrupt on Port 0?	`IsInterruptOnPort0()`
Interrupt on Port 1?	`IsInterruptOnPort1()`
Interrupt on Port 2?	`IsInterruptOnPort2()`
Interrupt on Port 3?	`IsInterruptOnPort3()`
Interrupt on Port 6?	`IsInterruptOnPort6()`
Receive Character via Serial Port	`ReceiveCharViaSerialPort(From Port)`
Receive N Characters via Serial Port	`ReceiveNCharsViaSerialPort(Put in, Num. Characters, From Port)`
Receive String via Serial Port	`ReceiveStringViaSerialPort(Put in, From Port)`
Receive Table via Serial Port	`ReceiveTableViaSerialPort(Start at Index, Of Table, From Port)`
Set End-of-Message Terminator	`SetEndOfMessageTerminator(To Character)`
Transmit Character via Serial Port	`TransCharViaSerialPort(Character, On Port)`
Transmit NewLine via Serial Port	`TransNewLineViaSerialPort(On Port)`
Transmit String via Serial Port	`TransStringViaSerialPort(String, On Port)`
Transmit Table via Serial Port	`TransTableViaSerialPort(Start at Index, Of Table, On Port)`
Transmit/Receive String via Serial Port	`TransReceStringViaSerialPort(String, On Port, Put Result in)`
Turn Off RTS	`TurnOffRts(On Port)`
Turn Off RTS After Next Character	`TurnOffRtsAfterNextChar()`
Turn On RTS	`TurnOnRts(On Port)`

1342-041206—OptoControl 4.1 Commands Quick Reference
Copyright © 1999–2004 Opto 22