Note: Commands with an asterisk* are available only in ioControl Professional.

	ioControl 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)`
	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 & 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 & Clear On-Latch	`GetClearOnLatch(From Point)`
	Get Counter	`GetCounter(From Point)`
	Get Frequency*	`GetFrequency(From Point)`
	Get Off-Latch	`GetOffLatch(From Point)`
	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	`GetOnLatch(From Point)`
	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)`
	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)`
	Stop Counter	`StopCounter(On Point)`
	Turn Off	`TurnOff(Output)`
	Turn On	`TurnOn(Output)`

	ioControl Command	OptoScript Equivalent (Arguments)
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)`
	Start Chart	`StartChart(Chart)`
	Stop Chart	`StopChart(Chart)`
	Suspend Chart	`SuspendChart(Chart)`
PID—Mistic	Clamp Mistic PID Output*	`ClampMisticPidOutput(High Clamp, Low Clamp, On PID Loop)`
	Clamp Mistic PID Setpoint*	`ClampMisticPidSetpoint(High Clamp, Low Clamp, On PID Loop)`
	Disable Mistic PID Output*	`DisableMisticPidOutput(Of PID Loop)`
	Disable Mistic PID Output Tracking in Manual Mode*	`DisableMisticPidOutputTrackingInManualMode(On PID Loop)`
	Disable Mistic PID Setpoint Tracking in Manual Mode*	`DisableMisticPidSetpointTrackingInManualMode(On PID Loop)`
	Enable Mistic PID Output*	`EnableMisticPidOutput(On PID Loop)`
	Enable Mistic PID Output Tracking in Manual Mode*	`EnableMisticPidOutputTrackingInManualMode(On PID Loop)`
	Enable Mistic PID Setpoint Tracking in Manual Mode*	`EnableMisticPidSetpointTrackingInManualMode(On PID Loop)`
	Get Mistic PID Control Word*	`GetMisticPidControlWord(From PID Loop)`
	Get Mistic PID D Term*	`GetMisticPidDTerm(From PID Loop)`
	Get Mistic PID I Term*	`GetMisticPidITerm(From PID Loop)`
	Get Mistic PID Input*	`GetMisticPidInput(PID Loop)`
	Get Mistic PID Mode*	`GetMisticPidMode(PID Loop)`
	Get Mistic PID Output*	`GetMisticPidOutput(PID Loop)`
	Get Mistic PID Output Rate of Change*	`GetMisticPidOutputRateOfChange(From PID Loop)`
	Get Mistic PID P Term*	`GetMisticPidPTerm(From PID Loop)`
	Get Mistic PID Scan Rate*	`GetMisticPidScanRate(From PID Loop)`
	Get Mistic PID Setpoint*	`GetMisticPidSetpoint(PID Loop)`
	Set Mistic PID Control Word*	`SetMisticPidControlWord(On-Mask, Off-Mask, For PID Loop)`
	Set Mistic PID D Term*	`SetMisticPidDTerm(To, On PID Loop)`
	Set Mistic PID I Term*	`SetMisticPidITerm(To, On PID Loop)`
	Set Mistic PID Input*	`SetMisticPidInput(PID Loop, Input)`
	Set Mistic PID Mode to Auto*	`SetMisticPidModeToAuto(On PID Loop)`
	Set Mistic PID Mode to Manual*	`SetMisticPidModeToManual(On PID Loop)`
	Set Mistic PID Output Rate of Change*	`SetMisticPidOutputRateOfChange(To, On PID Loop)`
	Set Mistic PID P Term*	`SetMisticPidPTerm(To, On PID Loop)`
	Set Mistic PID Scan Rate*	`SetMisticPidScanRate(To, On PID Loop)`
	Set Mistic PID Setpoint*	`SetMisticPidSetpoint(PID Loop, Setpoint)`

	ioControl Command	OptoScript Equivalent (Arguments)
High Density Digital	Clear HDD Module Off-Latches	`ClearHddModuleOffLatches(I/O Unit, Module Number, Clear Mask)`
	Clear HDD Module On-Latches	`ClearHddModuleOnLatches(I/O Unit, Module Number, Clear Mask)`
	Get & Clear All HDD Module Off-Latches	`GetClearAllHddModuleOffLatches(I/O Unit, Start Index, Put Result In)`
	Get & Clear All HDD Module On-Latches	`GetClearAllHddModuleOnLatches(I/O Unit, Start Index, Put Result In)`
	Get & Clear HDD Module Counter	`GetClearHddModuleCounter(I/O Unit, Module Number, Point Number, Put Result In)`
	Get & Clear HDD Module Counters	`GetClearHddModuleCounters(I/O Unit, Module Number, Start Table Index, Put Result In)`
	Get & Clear HDD Module Off-Latches	`GetClearHddModuleOffLatches(I/O Unit, Module Number, Put Result In)`
	Get & Clear HDD Module On-Latches	`GetClearHddModuleOnLatches(I/O Unit, Module Number, Put Result In)`
	Get All HDD Module Off-Latches	`GetAllHddModuleOffLatches(I/O Unit, Start Index, Put Result In)`
	Get All HDD Module On-Latches	`GetAllHddModuleOnLatches(I/O Unit, Start Index, Put Result In)`
	Get All HDD Module States	`GetAllHddModuleStates(I/O Unit, Start Index, Put Result In)`
	Get HDD Module Counters	`GetHddModuleCounters(I/O Unit, Module Number, Start Table Index, Put Result In)`
	Get HDD Module Off-Latches	`GetHddModuleOffLatches(I/O Unit, Module Number, Put Result In)`
	Get HDD Module On-Latches	`GetHddModuleOnLatches(I/O Unit, Module Number, Put Result In)`
	Get HDD Module States	`GetHddModuleStates(I/O Unit, Module Number, Put Result In)`
	Set HDD Module from MOMO Masks	`SetHddModulefromMOMOMasks(I/O Unit, Module Number, Must-On Mask, Must-Off Mask)`
	Turn Off HDD Module Point	`TurnOffHDDModulePoint(I/O Unit, Module Number, Point Number)`
	Turn On HDD Module Point	`TurnOnHddModulePoint(I/O Unit, Module Number, Point Number)`

1314-060308—ioControl Commands Quick Reference
Copyright © 2001–2006 Opto 22

	ioControl Command	OptoScript Equivalent (Arguments)
Error Handling	Add Message to Queue	`AddMessageToQueue(Severity, Message)`
	Add User Error to Queue	`AddUserErrorToQueue(Error Number)`
	Add User I/O Unit Error to Queue	`AddUserIoUnitErrorToQueue(Error Number, I/O Unit)`
	Caused a Chart Error?	`HasChartCausedError(Chart)`
	Caused an I/O Unit Error?	`HasIoUnitCausedError(I/O Unit)`
	Clear All Errors	`ClearAllErrors()`
	Copy Current Error to String	`CurrentErrorToString(Delimiter, String)`
	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 Error Code of Current Error	`GetErrorCodeOfCurrentError()`
	Get Error Count	`GetErrorCount()`
	Get ID of Block Causing Current Error	`GetIdOfBlockCausingCurrentError()`
	Get Line Causing Current Error	`GetLineCausingCurrentError()`
	Get Name of Chart Causing Current Error	`GetNameOfChartCausingCurrentError(Put in)`
	Get Name of I/O Unit Causing Current Error	`GetNameOfIoUnitCausingCurrentError(Put in)`
	Get Severity of Current Error	`GetSeverityOfCurrentError()`
	Remove Current Error and Point to Next Error	`RemoveCurrentError()`
	Stop Chart on Error	`StopChartOnError()`
	Suspend Chart on Error	`SuspendChartOnError()`
Control Engine	Calculate Strategy CRC	`CalcStrategyCrc()`
	Erase Files in Permanent Storage	`EraseFilesInPermanentStorage()`
	Get Available File Space	`GetAvailableFileSpace(File System Type)`
	Get Control Engine Address	`GetEngineAddress()`
	Get Control Engine Type	`GetEngineType()`
	Get Firmware Version	`GetFirmwareVersion(Put in)`
	Load Files From Permanent Storage	`LoadFilesFromPermanentStorage()`
	Retrieve Strategy CRC	`RetrieveStrategyCrc()`
	Save Files To Permanent Storage	`SaveFilesToPermanentStorage()`
I/O Unit—Event Msg	Get I/O Unit Event Message State	`GetIoUnitEventMsgState(I/O Unit, Event Message #, Put Result in)`
	Get I/O Unit Event Message Text	`GetIoUnitEventMsgText(I/O Unit, Event Message #, Put Result in)`
	Set I/O Unit Event Message State	`SetIoUnitEventMsgState(I/O Unit, Event Message #, State)`
	Set I/O Unit Event Message Text	`SetIoUnitEventMsgText(I/O Unit, Event Message #, Message Text)`

	ioControl 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 Maximum Value	`GetAnalogMaxValue(From)`
	Get Analog Minimum Value	`GetAnalogMinValue(From)`
	Get Analog Square Root Filtered Value*	`GetAnalogSquareRootFilteredValue(From)`
	Get Analog Sqaure Root Value*	`GetAnalogSquareRootValue(From)`
	Get Analog Totalizer Value*	`GetAnalogTotalizerValue(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 Load Cell Fast Settle Level	`SetAnalogLoadCellFastSettleLevel(To, On Point)`
	Set Analog Load Cell Filter Weight	`SetAnalogLoadCellFilterWeight(To, On Point)`
	Set Analog Totalizer Rate*	`SetAnalogTotalizerRate(To Seconds, On Point)`
	Set Analog TPO Period	`SetAnalogTpoPeriod(To, On Point)`
Time/Date	Copy Date to String (DD/MM/YYYY)	`DateToStringDDMMYYYY(String)`
	Copy Date to String (MM/DD/YYYY)	`DateToStringMMDDYYYY(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 Hours	`SetHours(To)`
	Set Minutes	`SetMinutes(To)`
	Set Month	`SetMonth(To)`
	Set Seconds	`SetSeconds(To)`
	Set Time	`SetTime(To)`
	Set Year	`SetYear(To)`

1314-060308—ioControl Commands Quick Reference
Copyright © 2001–2006 Opto 22

	ioControl 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 Event/Reaction*	`DisableCommunicationToEventReaction(`*`Event/Reaction`*`)`
	Disable Communication to I/O Unit	`DisableCommunicationToIoUnit(`*`I/O Unit`*`)`
	Disable Communication to Mistic PID Loop*	`DisableCommunicationtoMsiticPidLoop(`*`PID Loop`*`)`
	Disable Communication to PID Loop	`DisableCommunicationtoPidLoop(`*`PID Loop`*`)`
	Disable Communication to Point	`DisableCommunicationToPoint(`*`Point`*`)`
	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 Event/Reaction*	`EnableCommunicationToEventReaction(`*`Event/Reaction`*`)`
	Enable Communication to I/O Unit	`EnableCommunicationToIoUnit(`*`I/O Unit`*`)`
	Enable Communication to Mistic PID Loop*	`EnableCommunicationToMisticPidLoop(`*`PID Loop`*`)`
	Enable Communication to PID Loop	`EnableCommunicationtoPidLoop(`*`PID Loop`*`)`
	Enable Communication to Point	`EnableCommunicationToPoint(`*`Point`*`)`
	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 Point	`IvalSetAnalogPoint(`*`To, On Point`*`)`
	IVAL Set Counter	`IvalSetCounter(`*`To, On Point`*`)`
	IVAL Set Frequency*	`IvalSetFrequency(`*`To, On Point`*`)`
	IVAL Set Mistic PID Control Word*	`IvalSetPidControlWord(`*`On Mask, Off Mask, For PID Loop`*`)`
	IVAL Set Mistic PID Process Term*	`IvalSetMisticPidProcessTerm(`*`To, On PID Loop`*`)`
	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 TPO Percent*	`IvalSetTpoPercent(`*`To, On Point`*`)`
	IVAL Set TPO Period*	`IvalSetTpoPeriod(`*`Value, On Point`*`)`
	IVAL Turn Off	`IvalTurnOff(`*`Point`*`)`
	IVAL Turn On	`IvalTurnOn(`*`Point`*`)`
	Mistic PID Loop Communication Enabled?*	`IsMisticPidLoopCommEnabled(`*`PID Loop`*`)`
	PID Loop Communication Enabled?	`IsPidLoopCommEnabled(`*`PID Loop`*`)`

	ioControl Command	OptoScript Equivalent (Arguments)
PID—Ethernet	Get PID Configuration Flags	`GetPidConfigFlags(`*`PID Loop`*`)`
	Get PID Current Input	`GetPidCurrentInput(`*`PID Loop`*`)`
	Get PID Current Setpoint	`GetPidCurrentSetpoint(`*`PID Loop`*`)`
	Get PID Feed Forward	`GetPidFeedForward(`*`PID Loop`*`)`
	Get PID Feed Forward Gain	`GetPidFeedForwardGain(`*`PID Loop`*`)`
	Get PID Forced Output When Input Over Range	`GetPidForcedOutputWhenInputOverRange(`*`PID Loop`*`)`
	Get PID Forced Output When Input Under Range	`GetPidForcedOutputWhenInputUnderRange(`*`PID Loop`*`)`
	Get PID Gain	`GetPidGain(`*`PID Loop`*`)`
	Get PID Input	`GetPidInput(`*`PID Loop`*`)`
	Get PID Input High Range	`GetPidInputHighRange(`*`PID Loop`*`)`
	Get PID Input Low Range	`GetPidInputLowRange(`*`PID Loop`*`)`
	Get PID Max Output Change	`GetPidMaxOutputChange(`*`PID Loop`*`)`
	Get PID Min Output Change	`GetPidMinOutputChange(`*`PID Loop`*`)`
	Get PID Mode	`GetPidMode(`*`PID Loop`*`)`
	Get PID Output	`GetPidOutput(`*`PID Loop`*`)`
	Get PID Output High Clamp	`GetPidOutputHighClamp(`*`PID Loop`*`)`
	Get PID Output Low Clamp	`GetPidOutputLowClamp(`*`PID Loop`*`)`
	Get PID Scan Time	`GetPidScanTime(`*`PID Loop`*`)`
	Get PID Setpoint	`GetPidSetpoint(`*`PID Loop`*`)`
	Get PID Status Flags	`GetPidStatusFlags(`*`PID Loop`*`)`
	Get PID Tune Derivative	`GetPidTuneDerivative(`*`PID Loop`*`)`
	Get PID Tune Integral	`GetPidTuneIntegral(`*`PID Loop`*`)`
	Set PID Configuration Flags	`SetPidConfigFlags(`*`PID Loop, Configuration Flags`*`)`
	Set PID Feed Forward	`SetPidFeedForward(`*`PID Loop, Feed Forward`*`)`
	Set PID Feed Forward Gain	`SetPidFeedForwardGain(`*`PID Loop, Feed Fwd Gain`*`)`
	Set PID Forced Output When Input Over Range	`SetPidForcedOutputWhenInputOverRange(`*`PID Loop, Forced Output`*`)`
	Set PID Forced Output When Input Under Range	`SetPidForcedOutputWhenInputUnderRange(`*`PID Loop, Forced Output`*`)`
	Set PID Gain	`SetPidGain(`*`PID Loop, Gain`*`)`
	Set PID Input	`SetPidInput(`*`PID Loop, Input`*`)`
	Set PID Input High Range	`SetPidInputHighRange(`*`PID Loop, High Range`*`)`
	Set PID Input Low Range	`SetPidInputLowRange(`*`PID Loop, Low Range`*`)`
	Set PID Max Output Change	`SetPidMaxOutputChange(`*`PID Loop, Max Change`*`)`
	Set PID Min Output Change	`SetPidMinOutputChange(`*`PID Loop, Min Change`*`)`
	Set PID Mode	`SetPidMode(`*`PID Loop, Mode`*`)`
	Set PID Output	`SetPidOutput(`*`PID Loop, Output`*`)`
	Set PID Output High Clamp	`SetPidOutputHighClamp(`*`PID Loop, High Clamp`*`)`
	Set PID Output Low Clamp	`SetPidOutputLowClamp(`*`PID Loop, Low Clamp`*`)`
	Set PID Scan Time	`SetPidScanTime(`*`PID Loop, Scan Time`*`)`
	Set PID Setpoint	`SetPidSetpoint(`*`PID Loop, Setpoint`*`)`
	Set PID Tune Derivative	`SetPidTuneDerivative(`*`PID Loop, Derivative`*`)`
	Set PID Tune Integral	`SetPidTuneIntegral(`*`PID Loop, Integral`*`)`

1314-060308—ioControl Commands Quick Reference
Copyright © 2001–2006 Opto 22

	ioControl Command	OptoScript Equivalent (Arguments)
I/O Unit—Scratch Pad	Get I/O Unit Scratch Pad Bits	GetIoUnitScratchPadBits(*I/O Unit, Put Result in*)
	Get I/O Unit Scratch Pad Float Element	GetIoUnitScratchPadFloatElement(*I/O Unit, Index, Put Result in*)
	Get I/O Unit Scratch Pad Float Table	GetIoUnitScratchPadFloatTable(*I/O Unit, Length, From Index, To Index, To Table*)
	Get I/O Unit Scratch Pad Integer 32 Element	GetIoUnitScratchPadInt32Element(*I/O Unit, Index, Put Result in*)
	Get I/O Unit Scratch Pad Integer 32 Table	GetIoUnitScratchPadInt32Table(*I/O Unit, Length, From Index, To Index, To Table*)
	Get I/O Unit Scratch Pad String Element	GetIoUnitScratchPadStringElement(*I/O Unit, Index, Put Result in*)
	Get I/O Unit Scratch Pad String Table	GetIoUnitScratchPadString(*I/O Unit, Length, From Index, To Index, To Table*)
	Set I/O Unit Scratch Pad Bits from MOMO Mask	SetIoUnitScratchPadBitsFromMomo(*I/O Unit, Must-On Mask, Must-Off Mask*)
	Set I/O Unit Scratch Pad Float Element	SetIoUnitScratchPadFloatElement(*I/O Unit, Index, From*)
	Set I/O Unit Scratch Pad Float Table	SetIoUnitScratchPadFloatTable(*I/O Unit, Length, To Index, From Index, From Table*)
	Set I/O Unit Scratch Pad Integer 32 Element	SetIoUnitScratchPadInt32Element(*I/O Unit, Index, From*)
	Set I/O Unit Scratch Pad Integer 32 Table	SetIoUnitScratchPadInt32Table(*I/O Unit, Length, To Index, From Index, From Table*)
	Set I/O Unit Scratch Pad String Element	SetIoUnitScratchPadStringElement(*I/O Unit, Index, From*)
	Set I/O Unit Scratch Pad String Table	SetIoUnitScratchPadStringTable(*I/O Unit, Length, To Index, From Index, From Table*)

	ioControl Command	OptoScript Equivalent (Arguments)
Deprecated	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 Mixed I/O Unit from MOMO Masks	SetMixedIoUnitFromMomo(*Must-On Mask, Must-Off Mask, Mixed I/O Unit*)
	Set Mixed 64 I/O Unit from MOMO Masks	SetMixed64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Mixed 64 I/O Unit*)
	Set Simple 64 I/O Unit from MOMO Masks	SetSimple64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Simple 64 I/O Unit*)
	IVAL Set Digital Binary	IvalSetDigitalBinary(*On Mask, Off Mask, On I/O Unit*)
	IVAL Set Digital-64 I/O Unit from MOMO Masks	IvalSetDigital64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Digital 64 I/O Unit*)
	IVAL Set Mixed I/O Unit from MOMO Masks	IvalSetMixedIoUnitFromMomo(*Must-On Mask, Must-Off Mask, Mixed I/O Unit*)
	IVAL Set Mixed 64 I/O Unit from MOMO Masks	IvalSetMixed64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Mixed 64 I/O Unit*)
	IVAL Set Simple 64 I/O Unit from MOMO Masks	IvalSetSimple64IoUnitFromMomo(*Must-On Mask, Must-Off Mask, Simple 64 I/O Unit*)
I/O Unit	Get I/O Unit as Binary Value	GetIoUnitAsBinaryValue(*I/O Unit*)
	Get Target Address State*	GetTargetAddressState(*Enable Mask, Active Mask, I/O Unit*)
	I/O Unit Ready?	IsIoUnitReady(*I/O Unit*)
	IVAL Move Numeric Table to I/O Unit	IvalMoveNumTableToIoUnit(*Start at Index, Of Table, Move to*)
	Move I/O Unit to Numeric Table	MoveIoUnitToNumTable(*I/O Unit, Starting Index, Of Table*)
	Move Numeric Table to I/O Unit	MoveNumTableToIoUnit(*Start at Index, Of Table, Move to*)
	Set All Target Address States*	SetAllTargetAddressStates(*Must-On Mask, Must-Off Mask, Active Mask*)
	Set I/O Unit from MOMO Masks	SetIoUnitFromMomo(*Must-On Mask, Must-Off Mask, Digital I/O Unit*)
	Set Target Address State*	SetTargetAddressState(*Must-On Mask, Must-Off Mask, Active Mask, I/O Unit*)
	Write I/O Unit Configuration to EEPROM	WriteIoUnitConfigToEeprom(*On I/O Unit*)

Communication

ioControl Command	OptoScript Equivalent (Arguments)
Accept Incoming Communication	`AcceptIncomingCommunication(`*`Communication Handle`*`)`
Clear Communication Receive Buffer	`ClearCommunicationReceiveBuffer(`*`Communication Handle`*`)`
Clear Receive Buffer	`ClearReceiveBuffer`
Close Communication	`CloseCommunication(`*`Communication Handle`*`)`
Communication Open?	`IsCommunicationOpen(`*`Communication Handle`*`)`
Get Communication Handle Value	`GetCommunicationHandleValue(`*`From, To`*`)`
Get End-Of-Message Terminator	`GetEndOfMessageTerminator (`*`Communication Handle`*`)`
Get Number of Characters Waiting	`GetNumCharsWaiting(`*`On Communication Handle`*`)`
Listen for Incoming Communication	`ListenForIncomingCommunication(`*`Communication Handle`*`)`
Open Outgoing Communication	`OpenOutgoingCommunication(`*`Communication Handle`*`)`
Receive Character	`ReceiveChar(`*`Communication Handle`*`)`
Receive N Characters	`ReceiveNChars(`*`Put In, Number of Characters, Communication Handle`*`)`
Receive Numeric Table	`ReceiveNumTable(`*`Length, Start at Index, Of Table, Communication Handle`*`)`
Receive Pointer Table	`ReceivePtrTable(`*`Length, Start at Index, Of Table, Communication Handle`*`)`
Receive String	`ReceiveString(`*`Put In, Communication Handle`*`)`
Receive String Table	`ReceiveStrTable(`*`Length, Start at Index, Of Table, Communication Handle`*`)`
Send Communication Handle Command	`SendCommunicationHandleCommand(`*`Communication Handle, Command`*`)`
Set Communication Handle Value	`SetCommunicationHandleValue(`*`Value, Communication Handle`*`)`
Set End-Of-Message Terminator	`SetEndOfMessageTerminator (`*`Communication Handle, To Character`*`)`
Transfer N Characters	`TransferNChars(`*`Destination Handle, Source Handle, Num Chars`*`)`
Transmit Character	`TransmitChar(`*`Character, Communication Handle`*`)`
Transmit NewLine	`TransmitNewLine(`*`Communication Handle`*`)`
Transmit Numeric Table	`TransmitNumTable(`*`Length, Start at Index, Of Table, Communication Handle`*`)`
Transmit Pointer Table	`TransmitPtrTable(`*`Length, Start at Index, Of Table, Communication Handle`*`)`
Transmit/Receive Mistic I/O Hex String*	`TransReceMisticIoHexString(`*`Hex String, On Port, Put Result in`*`)`
Transmit/Receive String	`TransmitReceiveString(`*`String, Communication Handle, Put Result in`*`)`
Transmit String Table	`TransmitStrTable(`*`Length, Start at Index, Of Table, Communication Handle`*`)`
Transmit String	`TransmitString(`*`String, Communication Handle`*`)`

Mathematical

ioControl Command	OptoScript Equivalent (Arguments)
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`*`)`

1314-060308—ioControl Commands Quick Reference
Copyright © 2001–2006 Opto 22

	ioControl Command	OptoScript Equivalent (Arguments)
String	Append Character to String	`s1 += 'a';`
	Append String to String	`s1 += s2;`
	Compare Strings	`CompareStrings(String 1, String 2)`
	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 Integer 32 to IP Address String	`Int32ToIpAddressString(Convert, Put Result In)`
	Convert IP Address String to Integer 32	`IpAddressStringToInt32(Convert)`
	Convert Mistic I/O Hex String to Float*	`MisticIoHexToFloat(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 Mistic I/O Hex String*	`NumberToMisticIoHex(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 Element	`s = st[0];`
	Move String	`s1 = s2;`
	Move to String Table Element	`st[0] = s;`
	Move to String Table Elements	`MoveToStrTableElements(From, Start Index, End Index, Of Table)`
	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)`

	ioControl Command	OptoScript Equivalent (Arguments)
Miscellaneous	Comment (Block)	`/* block comment */`
	Comment (Single Line)	`// single line comment`
	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)`
	Get Type From Name	`GetTypeFromName(Name)`
	Get Value From Name	`GetValueFromName(Name, Put Result In)`
	Move	`x = y;`
	Move from Numeric Table Element	`x = nt[0];`
	Move Numeric Table Element to Numeric Table	`nt1[0] = nt2[5];`
	Move Numeric Table to Numeric Table	`MoveNumTableToNumTable(From Table, From Index, To Table, To Index, Length)`
	Move to Numeric Table Element	`nt[0] = x;`
	Move to Numeric Table Elements	`MoveToNumTableElements(From, Start Index, End Index, Of Table)`
	Shift Numeric Table Elements	`ShiftNumTableElements(Shift Count, Table)`
Pointer	Clear Pointer	`pn1 = null;`
	Clear Pointer Table Element	`pt[0] = null;`
	Get Pointer From Name	`GetPointerFromName(Name, Pointer)`
	Move from Pointer Table Element	`pn = pt[0];`
	Move to Pointer	`pn = &n;`
	Move to Pointer Table Element	`pt[0] = &n;`
	Pointer Equal to Null?	`pn == null`
	Pointer Table Element Equal to Null?	`pt[0] == null`
Event/Reaction	Clear All Event Latches*	`ClearAllEventLatches(On I/O Unit)`
	Clear Event Latch*	`ClearEventLatch(On 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 Scanning for All Events*	`EnableScanningForAllEvents(On I/O Unit)`
	Enable Scanning for Event*	`EnableScanningForEvent(Event/Reaction)`
	Enable Scanning of Event/Reaction Group*	`EnableScanningOfEventReactionGroup()`
	Event Occurred?*	`HasEventOccurred(Event/Reaction)`
	Event Occurring?*	`IsEventOccurring(Event/Reaction)`
	Event/Reaction Communication Enabled?*	`IsEventReactionCommEnabled(Event/Reaction)`
	Event Scanning Disabled?*	`IsEventScanningDisabled(Event/Reaction)`
	Event Scanning Enabled?*	`IsEventScanningEnabled(Event/Reaction)`
	Get & Clear Event Latches*	`GetClearEventLatches(E/R Group)`
	Get Event Latches*	`GetEventLatches(E/R Group)`
	Read Event/Reaction Hold Buffer*	`ReadEventReactionHoldBuffer(Event/Reaction)`

1314-060308—ioControl Commands Quick Reference
Copyright © 2001–2006 Opto 22

	ioControl Command	OptoScript Equivalent (Arguments)
Logical	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 Numeric 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 Numeric Table Element?	`x > nt[0]`
	Greater Than or Equal?	`x >= y`
	Greater Than or Equal to Numeric Table Element?	`x >= nt[0]`
	Less?	`x < y`
	Less Than Numeric Table Element?	`x < nt[0]`
	Less Than or Equal?	`x <= y`
	Less Than or Equal to Numeric 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 Numeric Table Element?	`n <> nt[0]`
	Numeric Table Element Bit Clear	`NumTableElementBitClear(Element Index, Of Integer Table, Bit to Clear)`
	Numeric Table Element Bit Set	`NumTableElementBitSet(Element Index, Of Integer Table, Bit to Set)`
	Numeric Table Element Bit Test	`NumTableElementBitTest(Element Index, Of Integer Table, Bit to Test)`
	OR	`x or y`
	OR?	*See* OR
	Set Variable False	`SetVariableFalse(Variable)`
	Set Variable True	`SetVariableTrue(Variable)`
	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?

	ioControl Command	OptoScript Equivalent (Arguments)
	Test Not Equal	*See* Not Equal?
	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
	Pointer Table Element Equal to Null?	`pt[0] == null`
I/O Unit—Memory Map	Read Number from I/O Unit Memory Map	`ReadNumFromIoUnitMemMap(I/O Unit, Mem address, To)`
	Read Numeric Table from I/O Unit Memory Map	`ReadNumTableFromIoUnitMemMap(Length, Start Index, I/O Unit, Mem address, To)`
	Read String from I/O Unit Memory Map	`ReadStrFromIoUnitMemMap(Length, I/O Unit, Mem address, To)`
	Read String Table from I/O Unit Memory Map	`ReadStrTableFromIoUnitMemMap(Length, Start Index, I/O Unit, Mem address, To)`
	Write Number to I/O Unit Memory Map	`WriteNumToIoUnitMemMap(I/O Unit, Mem address, Variable)`
	Write Numeric Table to I/O Unit Memory Map	`WriteNumTableToIoUnitMemMap(Length, Start Index, I/O Unit, Mem address, Table)`
	Write String to I/O Unit Memory Map	`WriteStrToIoUnitMemMap(I/O Unit, Mem address, Variable)`
	Write String Table to I/O Unit Memory Map	`WriteStrTableToIoUnitMemMap(Length, Start Index, I/O Unit, Mem address, Table)`
Timing	Continue Timer	`ContinueTimer(Timer)`
	Delay (mSec)	`DelayMsec(Milliseconds)`
	Delay (Sec)	`DelaySec(Seconds)`
	Down Timer Expired?	`HasDownTimerExpired(Down Timer)`
	Pause Timer	`PauseTimer(Timer)`
	Set Down Timer Preset Value	`SetDownTimerPreset(Target Value, Down Timer)`
	Set Up Timer Target Value	`SetUpTimerTarget(Target Value, Up Timer)`
	Start Timer	`StartTimer(Timer)`
	Stop Timer	`StopTimer(Timer)`
	Timer Expired?	`HasTimerExpired(Timer)`
	Up Timer Target Time Reached?	`HasUpTimerReachedTargetTime(Up Timer)`

1314-060308—ioControl Commands Quick Reference
Copyright © 2001–2006 Opto 22