

This is Google's cache of <http://www.openservo.com/TWIProtocol>. It is a snapshot of the page as it appeared on 6 Sep 2014 16:47:56 GMT. The [current page](#) could have changed in the meantime. [Learn more](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.

[Text-only version](#)

## [MoinMoin Logo](#)

- [Login](#)
- [TWIProtocol](#)
- [RecentChanges](#)
- [FindPage](#)
- [HelpContents](#)
- [TWIProtocol](#)
- Immutable Page
- [Info](#)
- [Attachments](#)
- More Actions:



# OpenServo TWI/I2C Protocol

**For the Version 3 development branch layout see [TWIProtocol-dev](#)**

NOTE: I2C (pronounced "eye-square-see") is the two-wire bidirectional protocol utilized by the OpenServo for communication with a master controller. The protocol was invented by Philips and they have protected this name. For this reason other manufacturers have adopted different names for the same protocol. Atmel calls I2C "two wire interface" (TWI). This document adopts the TWI abbreviation used by Atmel for the bidirectional communication protocol to the OpenServo.

## OpenServo Data Registers

The OpenServo implements 48 status registers that can be read from a master controller to modify and monitor the state of the servo. These registers are read using the TWI-bus DATA READ cycle and written using the TWI-bus DATA WRITE cycle.

NOTE: The OpenServo source code is the best documentation regarding the definition of each data register. Please refer to the 'registers.h' file for specific definitions of each register.

### Read Only Registers

Registers 0x00 thru 0x0F are read-only and cannot be modified by the master controller. These registers implement servo status values and flags that can be modified by commands.

### Read/Write Registers

Registers 0x10 thru 0x1F are always read-write enabled and can be modified by the master controller. These values implement dynamic servo control such as positioning.

### Read/Write Protected Registers

Registers 0x20 thru 0x2F are write protected in that they can only be written by the master controller when the write enable flag is set via the WRITE\_ENABLE command. These registers implement configuration information and are normally only written to during initial configuration of the servo.

### OpenServo Register Table

Address Name		Type	Description
0x00	DEVICE_TYPE	Read Only	Device type - 1 = OpenServo device type
0x01	DEVICE_SUBTYPE	Read Only	Device subtype - 1 = OpenServo device subtype
0x02	VERSION_MAJOR	Read Only	Major version number of OpenServo software
0x03	VERSION_MINOR	Read Only	Minor version number of OpenServo software

0x04	FLAGS_HI	Read Only	Flags high byte
0x05	FLAGS_LO	Read Only	Flags low byte
0x06	TIMER_HI	Read Only	Timer high byte - incremented each ADC sample
0x07	TIMER_LO	Read Only	Timer low byte - incremented each ADC sample
0x08	POSITION_HI	Read Only	Servo position high byte
0x09	POSITION_LO	Read Only	Servo position low byte
0x0A	VELOCITY_HI	Read Only	Servo velocity high byte
0x0B	VELOCITY_LO	Read Only	Servo velocity low byte
0x0C	POWER_HI	Read Only	Servo power high byte
0x0D	POWER_LO	Read Only	Servo power low byte
0x0E	PWM_CW	Read Only	PWM clockwise value
0x0F	PWM_CCW	Read Only	PWM counter-clockwise value
0x10	SEEK_HI	Read/Write	Seek position high byte
0x11	SEEK_LO	Read/Write	Seek position low byte

0x12	SEEK_VELOCITY_HI	Read/Write	Speed seek position high byte
0x13	SEEK_VELOCITY_LO	Read/Write	Speed seek position low byte
0x14	VOLTAGE_HI	Read/Write	Battery Voltage value high byte
0x15	VOLTAGE_LO	Read/Write	Battery Voltage value low byte
0x16	CURVE_RESERVED	Read/Write	reserved curve data
0x17	CURVE_BUFFER	Read/Write	Remaining curve buffer space
0x18	CURVE_DELTA_HI	Read/Write	Curve Time delta high byte
0x19	CURVE_DELTA_LO	Read/Write	Curve Time delta low byte
0x1A	CURVE_POSITION_HI	Read/Write	Curve position high byte
0x1B	CURVE_POSITION_LO	Read/Write	Curve position low byte
0x1C	CURVE_IN_VELOCITY_HI	Read/Write	Curve in velocity high byte
0x1D	CURVE_IN_VELOCITY_LO	Read/Write	Curve in velocity low byte
0x1E	CURVE_OUT_VELOCITY_HI	Read/Write	Curve out velocity high byte

# OpenServo Commands

The OpenServo implements the following commands that can be sent by the master controller to enable certain operations of the servo. These commands are sent to the OpenServo using the TWI-bus COMMAND cycle.

NOTE: Please refer to the 'twi.h' file for specific definitions of each command.

Command Name		Description
0x00...0x7F reserved		Reserved for data addresses (msb = 0)
0x80	RESET	Reset microcontroller
0x81	CHECKED_TXN	Read/Write registers with simple checksum
0x82	PWM_ENABLE	Enable PWM to motors
0x83	PWM_DISABLE	Disable PWM to servo motors
0x84	WRITE_ENABLE	Enable write of read/write protected registers
0x85	WRITE_DISABLE	Disable write of read/write protected registers
0x86	REGISTERS_SAVE	Save read/write protected registers to EEPROM
0x87	REGISTERS_RESTORE	Restore read/write protected registers from EEPROM
0x88	REGISTERS_DEFAULT	Restore read/write protected registers to defaults

# OpenServo TWI-bus Protocol

Before any data is transmitted on the TWI-bus, the slave device that should respond is addressed first. Slave addressing is always carried out with the first byte transmitted after the start condition. The TWI-bus configuration for the COMMAND, DATA WRITE and DATA READ cycles is shown below.

## TWI-bus COMMAND Cycle

command\_cycle.gif

After the slave address with the read/write flag set to 0, an unlimited number of command bytes can be sent. Each command byte must have the address/command flag (most significant bit) set to 1. A DATA WRITE/READ cycle can be entered by sending an address with the most significant bit set to 0.

## TWI-bus DATA WRITE Cycle

data\_write\_cycle2.gif

After the slave address with the read/write flag set to 0, a data address is written with the address/command flag (most significant bit) set to 0. The address is latched. An unlimited number of data bytes can be written in one operation with the address being automatically incremented after each write. The address will wrap to 0x00 after address 0x7F is written. Data writing will continue until a TWI stop or start condition is set by the master.

## TWI-bus DATA READ Cycle

data\_read\_cycle2.gif

After the slave address with the read/write flag set to 0, a data address is written with the address/command flag (most significant bit) set to 0. The address is latched. To read each data byte the master then sets a start condition and sends the slave address with the read/write flag set to 1 indicating data reads will follow. An unlimited number of data bytes can be read in one operation with the address being automatically incremented after each read. The address will wrap to 0x00 after address 0x7F is read. Data reading will continue until a TWI stop or start condition is set by the master. For correct operation, the master should negative acknowledge the last byte read before setting a stop or start condition.

NOTE: It may be advantageous to set a start condition, prior to reading from the openservo, by issuing a "restart", rather than "stop" followed by "start"- this will

prevent another TWI master or device from taking control of the bus.

## OpenServo TWI-bus Checked Protocol

**TBD**

Return to [Docs, Tools & Utilities](#)

TWIProtocol (last edited 2012-10-12 19:51:18 by localhost)

- Immutable Page
- [Info](#)
- [Attachments](#)

• More Actions:

- [MoinMoin Powered](#)
- [Python Powered](#)
- [GPL licensed](#)
- [Valid HTML 4.01](#)