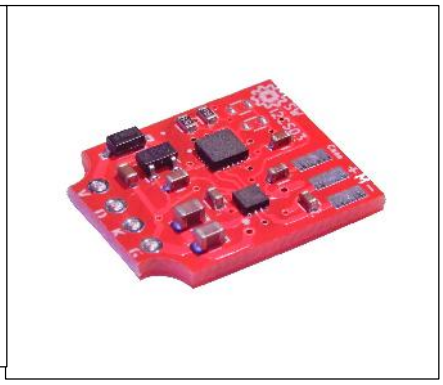
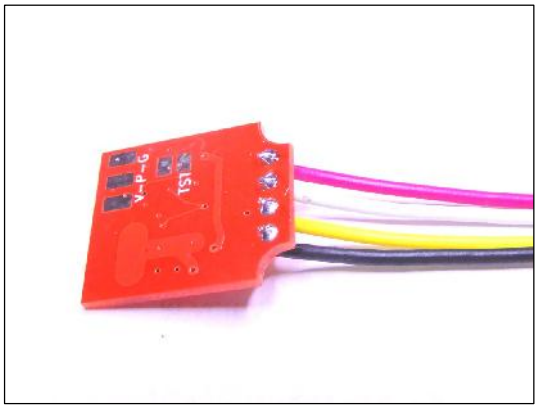
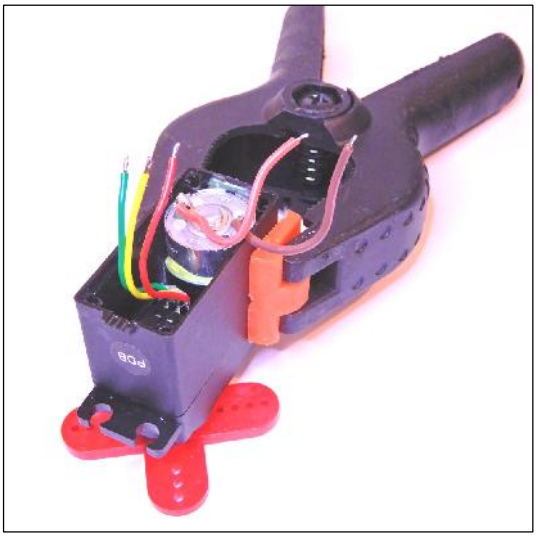
SWI2CS003
I2C Servo Board (Fits standard 3 to 6Kgm Servo)

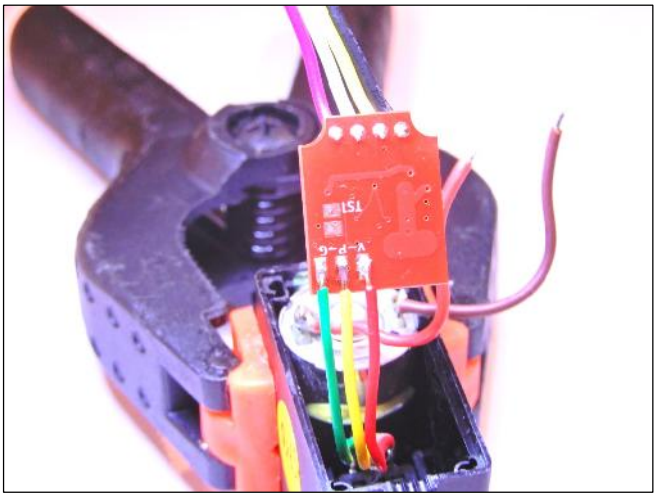


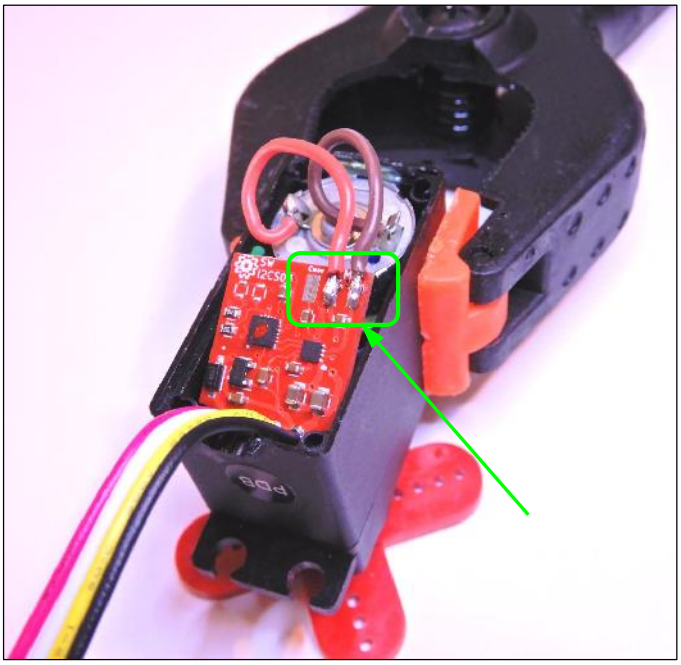
APPLICATION DIAGRAM

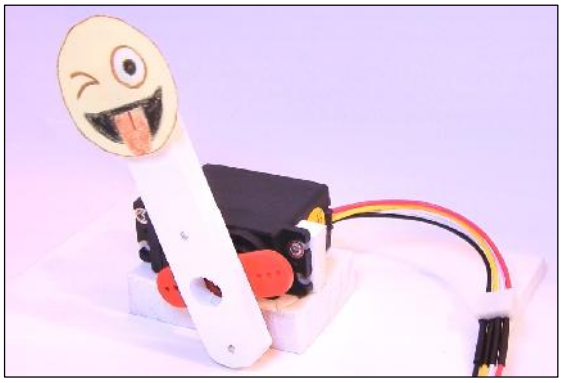
SETUP











I2C COMMUNICATION

Address assignment

REGISTER SPACE

OFFSET	NAME	R/W	RANGE	DEFAULT	DESCRIPTION
0x00	FIRMWARE_VERSION	R	1-N	1	Current software revision
0x01	I2C_ADDRESS	R/W	0-127	0	Bus address. Written new address is saved immediately Further access must be to the new address
0x02	STOP	W	1	0	Stop Immediately with no ramping, and enter park state
0x03	PARK_TYPE	R/W	0-255	1	0 = Coast, free motor. No power consumption. 1 = Brake, driver ties up motor leads. No power consumption. 2 = Active hold with compliance with power = MAX_POWER 3-100 = Active hold with compliance. Power: From this number
0x04	MOTOR_POLARITY	R/W	0-1	1	Used as correction if wires are soldered backwards to motor 0=CCW 1=CW
0x05	CONTINUOUS ROTATION	R/W	0-1	0	0 = single turn, normal servo operation. Uses pot range 0-1024 1 = Continuous rotation with MAX_POWER as speed control
0x06	MAX_POWER	R/W	0-100	80	0 % = no power 100% = Maximum power when seeking or holding position
0x07	MAX_SPEED_H	R/W	0-5000	2000	Ramp profile speed limit. Actual speed reached will depend on servo type and load Typically 2000 units per second for 0.16s-60deg servo
0x08	MAX_SPEED_L	R/W			
0x09	RAMP_TIME_H	R/W	1-10000	250	mSeconds to reach MAX_SPEED (if power and load permit)
0x0A	RAMP_TIME_L				
0x0B	RAMP_CURVE	R/W	0-100	50	0 = Linear ramp profile 100 = Sine ramp profile
0x0C	DEADBAND WINDOW	R/W	0-20	1	Target position tolerance. Helps with flexibility
0x0D	TARGET_POSITION_H	R/W	0-1024	0	Write desired position, this will override any current motion.
0x0E	TARGET_POSITION_L	R/W			
0x0F	TARGET_POSITION_NEXT	R/W	0-1024		Push new position here without having to wait for current motion completion.
0x10	TARGET_POSITION_NEXT	R/W			
0x0F	CURRENT_STATE	R	0-2		Operation State. 0:Parked 1:seeking 2:Running stored program

0x10	CURRENT_POSITION_H	R/W	0-1024		Current position
0x11	CURRENT_POSITION_L	R/W			
0x12	CURRENT_VELOCITY_H	R	- 5000 +5000		Current velocity in Position units/Sec
0x13	CURRENT_VELOCITY_L	R			
0x14	CURRENT_POWER	R	0-100		Current power % being applied to exert motion or hold
0x15	CURRENT_TEMPERATURE	R	0-100		CPU temperature in Celsius
0x16	LAST_CRC8	R	0-255		When using i2c crc scheme. See example projects.
0x17	PROGRAM_POSITION_N_H	R/W	0-1024		(N) x8 slots.. .. 0=unused slot/end of sequence
0x18	PROGRAM_POSITION_N_L	R/W			
0x28	PROGRAM_REPS	R/W	0-255		Reps to do programmed sequence upon power up. 255=inf
0x2A	SAVE	W	1	0	Saves all registers. Motion settings loaded on next power up.

Register Detailed Description

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

0x00 FIRMWARE VERSION R Range: 1-N Default

Current software version

EXAMPLES

The following pseudo code shows some basic read and write cases.
For full example projects and library. Go to github.statorworks/i2cServo/examples

Address assignment

Raw:
```
I2c_write(address, new_address, REG_I2C_ADDRESS);
```

Using library:
```
I2CServo_SetAddress(address, new_address, 0);
```

Set Max speed

Set target position

Read target position

Set target position, with CRC8 transmission checksum