Angular Displacement Sensor (ADS) I2C Protocol Specification

1. General Specifications

- Default I2C Address: 0x12
- Bootloader Address: 0x12
- I2C Speed: 400 kHz
- Packet Size: 3 bytes

2. I2C Transaction Format

All I2C transactions use a standard 3-byte packet format.

2.1 Write Format

- Byte 0: Command code
- Bytes 1-2: Command parameters

2.2 Read Format

- Byte 0: Packet type
- Bytes 1-2: Data (typically encoded as a 16-bit value)

3. Command Codes

Command Code (Hex)	Command Name	Description
0x00	RUN	Set sensor in free-run or standby mode
0x01	SPS	Set sample rate for free-run mode
0x02	RESET	Software reset
0x03	DFU	Enter bootloader mode for firmware update
0x04	SET_ADDRESS	Update the I2C address
0x05	POLLED_MODE	Set sensor in polled mode or standby

Command Code (Hex)	Command Name	Description
0x06	GET_FW_VER	Get firmware version
0x07	CALIBRATE	Perform sensor calibration
0x08	READ_STRETCH	Enable/disable stretch measurements
0x09	SHUTDOWN	Enter ultra-low power mode (~50nA)
0x0A	GET_DEV_ID	Get device ID

4. Packet Types (Response Codes)

Packet Type (Hex)	Name	Description
0x00	SAMPLE	Packet contains bend angle data
0x01	FW_VER	Packet contains firmware version
0x02	DEV_ID	Packet contains device ID
0x03	STRETCH_SAMPLE	Packet contains stretch data

5. Command Details

5.1 RUN (0x00)

Sets the sensor in free-run interrupt mode or standby.

Write Format:

- Byte 0: 0x00
- Byte 1: 0x01 = activate, 0x00 = standby
- Byte 2: 0x00 (unused)

Example:

```
To activate: 0x00 0x01 0x00
To standby:  0x00 0x00 0x00
```

5.2 SPS (0x01)

Sets the sample rate in free-run mode.

Write Format:

- Byte 0: 0x01

- Bytes 1-2: Sample rate value (16-bit, little-endian)

Sample Rate Values:

Value (Hex)	Value (Dec)	Rate	Notes
0x4000	16384	1 Hz	
0x0666	1638	10 Hz	
0x0333	819	20 Hz	
0x0147	327	50 Hz	
0x00A3	163	100 Hz	
0x0051	81	200 Hz	Max rate for bend + stretch
0x0031	49	333 Hz	
0x0020	32	500 Hz	Max rate

Example:

```
For 100 Hz: 0x01 0xA3 0x00
```

5.3 RESET (0x02)

Performs a software reset.

Write Format:

- Byte 0: 0x02
- Bytes 1-2: 0x00 0x00 (unused)

Example:

```
0x02 0x00 0x00
```

5.4 DFU (0x03)

Resets the sensor into bootloader mode for firmware update.

Write Format:

- Byte 0: 0x03
- Bytes 1-2: 0x00 0x00 (unused)

Example:

```
0x03 0x00 0x00
```

5.5 SET_ADDRESS (0x04)

Updates the I2C address of the sensor.

Write Format:

- Byte 0: 0x04
- Byte 1: New 7-bit I2C address
- Byte 2: 0x00 (unused)

Example:

```
To set address 0x13: 0x04 0x13 0x00
```

5.6 POLLED_MODE (0x05)

Places sensor in polled mode or standby. In polled mode, a new sample is taken each time sensor data is read.

Write Format:

- Byte 0: 0x05
- Byte 1: 0x01 = activate, 0x00 = standby
- Byte 2: 0x00 (unused)

Example:

```
To activate: 0x05 0x01 0x00
To standby:  0x05 0x00 0x00
```

5.7 GET_FW_VER (0x06)

Gets the firmware version.

Write Format:

- Byte 0: 0x06
- Bytes 1-2: 0x00 0x00 (unused)

Read Response:

- Byte 0: 0x01 (FW_VER)
- Bytes 1-2: Firmware version (16-bit, little-endian)

Example:

```
Write: 0x06 0x00 0x00
Read response: 0x01 0x2A 0x01 (firmware version 0x012A = 298)
```

5.8 CALIBRATE (0x07)

Performs sensor calibration.

Write Format:

- Byte 0: 0x07
- Byte 1: Calibration step (see table below)
- Byte 2: Angle in degrees (for step 0x00 and 0x01) or 0x00 for other steps

Calibration Steps:

Value (Hex)	Description
0x00	First calibration point (typically 0°)
0x01	Second calibration point (45°-255°)
0x02	Clear user calibration, restore factory
0x03	0mm strain calibration point
0x04	Second strain point (typically 30mm)

Example:

```
For 0° calibration: 0x07 0x00 0x00
For 90° calibration: 0x07 0x01 0x5A  (0x5A = 90 in decimal)
```

5.9 READ_STRETCH (0x08)

Enables or disables stretch measurements.

Write Format:

- Byte 0: 0x08
- Byte 1: 0x01 = enable, 0x00 = disable
- Byte 2: 0x00 (unused)

Example:

```
To enable: 0x08 0x01 0x00
To disable: 0x08 0x00 0x00
```

5.10 SHUTDOWN (0x09)

Puts the sensor in ultra-low power mode (~50nA). Requires reset to wake up.

Write Format:

- Byte 0: 0x09
- Bytes 1-2: 0x00 0x00 (unused)

Example:

```
0x09 0x00 0x00
```

5.11 GET_DEV_ID (0x0A)

Gets the device ID.

Write Format:

- Byte 0: 0x0A
- Bytes 1-2: 0x00 0x00 (unused)

Read Response:

- Byte 0: 0x02 (DEV_ID)
- Byte 1: Device ID (see table below)
- Byte 2: 0x00 (unused)

Device ID Values:

Value (Hex)	Description
0x01	One-axis sensor v1
0x02	Two-axis sensor v1
0x0C	One-axis sensor v2

Example:

```
Write: 0x0A 0x00 0x00
Read response: 0x02 0x01 0x00 (One-axis sensor v1)
```

6. Data Interpretation

6.1 Bend Angle Data

When reading from the sensor in any mode, a packet with type 0x00 (SAMPLE) contains bend angle data.

Read Format:

- Byte 0: 0x00 (SAMPLE)
- Bytes 1-2: Bend angle data (16-bit signed int, little-endian)

Conversion:
Angle in degrees = (16-bit value) / 64.0

Example:

```
Read: 0x00 0x80 0x01
16-bit value: 0x0180 = 384
Angle = 384 / 64.0 = 6.0 degrees
```

6.2 Stretch Data

When stretch measurements are enabled, packets with type 0x03 (STRETCH_SAMPLE) contain stretch data.

Read Format:

- Byte 0: 0x03 (STRETCH_SAMPLE)
- Bytes 1-2: Stretch data (16-bit signed int, little-endian)

Conversion:
Stretch value = (16-bit value) / 64.0

Example:

```
Read: 0x03 0x00 0x05
16-bit value: 0x0500 = 1280
Stretch = 1280 / 64.0 = 20.0
```

7. Typical I2C Operation Examples

7.1 Initialize in Free-Run Mode (100 Hz)

```
# Set sample rate to 100 Hz
Write: 0x01 0xA3 0x00

# Start sampling in free-run mode
Write: 0x00 0x01 0x00

# Sensor will now generate samples at 100 Hz on the interrupt pin
```

7.2 Initialize in Polled Mode

```
# Enter polled mode
Write: 0x05 0x01 0x00
```

```
# Read a sample when needed
Read: 0x00 0x80 0x01 (example response with bend data)
```

7.3 Calibration Sequence

```
# First calibration point at 0 degrees
Write: 0x07 0x00 0x00

# Second calibration point at 90 degrees
Write: 0x07 0x01 0x5A
```

7.4 Enable Stretch and Bend Measurements

```
# Enable stretch measurements
Write: 0x08 0x01 0x00

# Sensor will now provide both bend (type 0x00) and stretch (type 0x03) packets
```

8. Bootloader Operation

8.1 Update Process

1. Reset the device into bootloader mode:

   ```
   Write: 0x03 0x00 0x00
   ```

2. Subsequent communications should be addressed to the bootloader address (0x12) and follow the bootloader protocol.

3. The bootloader protocol involves:

   - Sending a 4-byte firmware length
   - Sending firmware data in 64-byte pages
   - Receiving acknowledgments between operations

Note: Detailed bootloader protocol is available in the firmware update documentation.