BNO055 Custom Axis Remap Configuration

Problem with Adafruit Library

The Adafruit BNO055 library only provides 8 preset configurations (P0-P7), but these are limited:

- P0-P7 only use values **0x21** and **0x24**
- Your X↔Z swap requires **0x06** (not available in presets!)

Solution: Direct Register Writing

Write custom values directly to BNO055 registers, bypassing the library presets.

Register Details from Datasheet

AXIS_MAP_CONFIG Register (0x41)

Controls which physical axis maps to which software axis:

Bits	Function	Values
5-4	Remapped Z axis	00=X, 01=Y, 10=Z
3-2	Remapped Y axis	00=X, 01=Y, 10=Z
1-0	Remapped X axis	00=X, 01=Y, 10=Z

Important: "Remapped Z axis" means "where does SOFTWARE Z get its data from?"

AXIS_MAP_SIGN Register (0x42)

Controls the sign (positive/negative) of each axis:

Bit	Function	Values
2	Z axis sign	0=positive, 1=negative
1	Y axis sign	0=positive, 1=negative
0	X axis sign	0=positive, 1=negative

Your Configuration: X↔Z Swap

Goal:

Goal:

- Physical X → Software Z
- Physical Y → Software Y (unchanged)
- Physical Z → Software X

AXIS_MAP_CONFIG calculation:

- Software Z gets Physical X → bits 5-4 = **00** (X-Axis)
- Software Y gets Physical Y → bits 3-2 = **01** (Y-Axis)
- Software X gets Physical Z → bits 1-0 = **10** (Z-Axis)

Binary: 00 01 10 = **0x06**

AXIS_MAP_SIGN: Start with **0x00** (all positive), adjust if needed.

Implementation

In the sketch, at the top:

```cpp
// X↔Z swap with Y unchanged
const uint8_t CUSTOM_AXIS_REMAP_CONFIG = 0x06;

// All axes positive (adjust if signs wrong)
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x00;
```

Testing and Adjustment

Step 1: Upload and Test

1. Upload `BNO055_Custom_Axis_Remap.ino`
2. Hold sensor with physical X+ pointing UP (vertical)
3. Check Serial Monitor output

Step 2: Check Axis Alignment

Expected (correct):

```
Software Z ≈ +9.8 m/s²
Software X ≈ 0
Software Y ≈ 0
```

```
Orientation: VERTICAL (X+ up) - Normal
```

If Z is negative instead:

```
Software Z ≈ -9.8 m/s² ← Wrong sign!
```

Step 3: Adjust Signs if Needed

If any axis has the wrong sign, modify `CUSTOM_AXIS_REMAP_SIGN`:

```cpp
// Sign adjustment options
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x00;  // Binary: 000 - all positive
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x01;  // Binary: 001 - flip X
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x02;  // Binary: 010 - flip Y
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x04;  // Binary: 100 - flip Z
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x05;  // Binary: 101 - flip Z and X
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x07;  // Binary: 111 - flip all
```

Most common: If Z shows -9.8 instead of +9.8, use **0x04** to flip Z.

Other Custom Configurations

You can create ANY mapping you need! Here are some examples:

Example 1: Y↔Z Swap (X unchanged)

```cpp
// Software Z gets Physical Y (01), Y gets Physical Z (10), X gets Physical X (00)
// Binary: 01 10 00 = 0x18
const uint8_t CUSTOM_AXIS_REMAP_CONFIG = 0x18;
```

Example 2: Cyclic Rotation (X→Y→Z→X)

```cpp
// Software Z gets Physical Y (01), Y gets Physical X (00), X gets Physical Z (10)
// Binary: 01 00 10 = 0x12
const uint8_t CUSTOM_AXIS_REMAP_CONFIG = 0x12;
```

Example 3: All Inverted

```cpp
```

```
// Same mapping but negative signs
const uint8_t CUSTOM_AXIS_REMAP_CONFIG = 0x06;  // X↔Z swap
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x07;    // All negative
```

Calculation Tool

To calculate any custom configuration:

1. **Decide the mapping:**
    - Where should Software X get data from?
    - Where should Software Y get data from?
    - Where should Software Z get data from?
2. **Convert to bits:**
    - Physical X = 00
    - Physical Y = 01
    - Physical Z = 10
3. **Assemble:**
    - Bits 5-4 = Software Z source
    - Bits 3-2 = Software Y source
    - Bits 1-0 = Software X source
4. **Example:**
    - Software X from Physical Y → bits 1-0 = 01
    - Software Y from Physical Z → bits 3-2 = 10
    - Software Z from Physical X → bits 5-4 = 00
    - Result: 00 10 01 = 0x09

Verification

The sketch includes automatic verification:

```
Verification - Config: 0x06 ✓
Verification - Sign: 0x00 ✓
```

If you see ✗ MISMATCH, the register write failed (rare).

Why This Works Better

Adafruit Presets:

- Limited to 0x21 and 0x24
- Only 8 configurations
- Your X↔Z swap not available

Custom Direct Write:

- Any value 0x00-0x3F possible
- Full control over mapping
- Exactly what you need!

Complete Mapping Table

All possible values for each axis:

Bits	Value	Physical Source
00	0	X-Axis
01	1	Y-Axis
10	2	Z-Axis
11	3	Invalid

Troubleshooting

"Verification shows MISMATCH"

Cause: Register write didn't take **Solution:**

- Check sensor is in CONFIG mode (handled automatically)
- Increase delay after write
- Try power cycle

"Z is correct value but negative"

Cause: Sign bit needs flipping **Solution:** Change `CUSTOM_AXIS_REMAP_SIGN` to 0x04

"Values still wrong"

Cause: Wrong config value calculated **Solution:**

1. Check which axis shows 9.8 when X+ is physically up
2. Recalculate based on actual vs desired mapping

"All axes seem rotated wrong"

Cause: Might have calculated inverse mapping **Solution:** Try swapping the bit positions

Summary

For your **X↔Z swap with Y unchanged**:

```cpp
const uint8_t CUSTOM_AXIS_REMAP_CONFIG = 0x06;  // Binary: 00 01 10
const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x00;    // Binary: 000

// If Z is negative, change sign to:
// const uint8_t CUSTOM_AXIS_REMAP_SIGN = 0x04;  // Binary: 100 (flip Z)
```

This bypasses the limited Adafruit presets and gives you **exactly** the mapping you need! 🎯