Sensors

ST6002CEM – Mobile Application Development

What is a Sensor?

- A Sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena.
- The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.

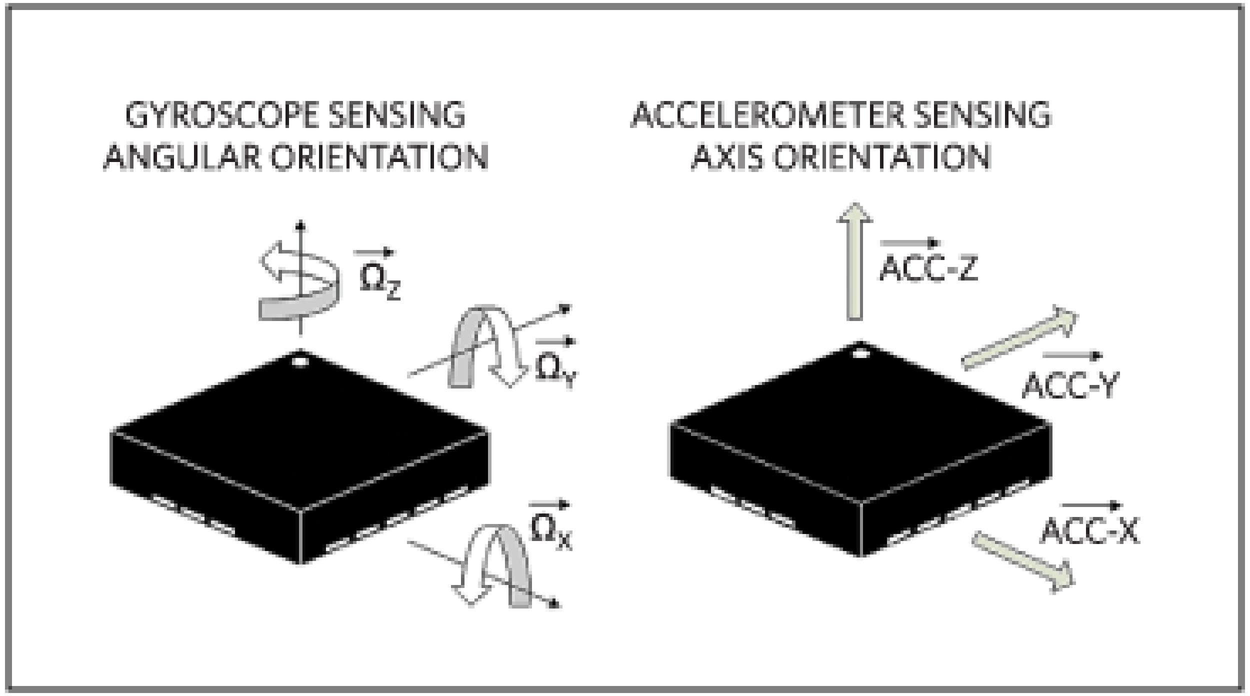
Various types of sensors in Smartphone

1. An accelerometer detects acceleration, tilt, and vibration to determine movement and orientation.
2. A gyroscope identifies up/down, left/right, and rotation around three axes for more complex orientation details.
3. A light sensor detects data about lighting levels in the environment to adapt the display accordingly.
4. A proximity sensor detects when the phone is held to the face to make or take a call, so the touchscreen display can be disabled to avoid unintended input.
5. A fingerprint sensor can enable biometric verification for secure device and website authentication as well as mobile payment.
6. A magnetometer detects the direction of magnetic north and, in conjunction with GPS, determines the user's location.
7. An infrared sensor can be used to identify user movements for gesture recognition.

Download starter project

https://github.com/kiranrana8973/sensor_starter.git

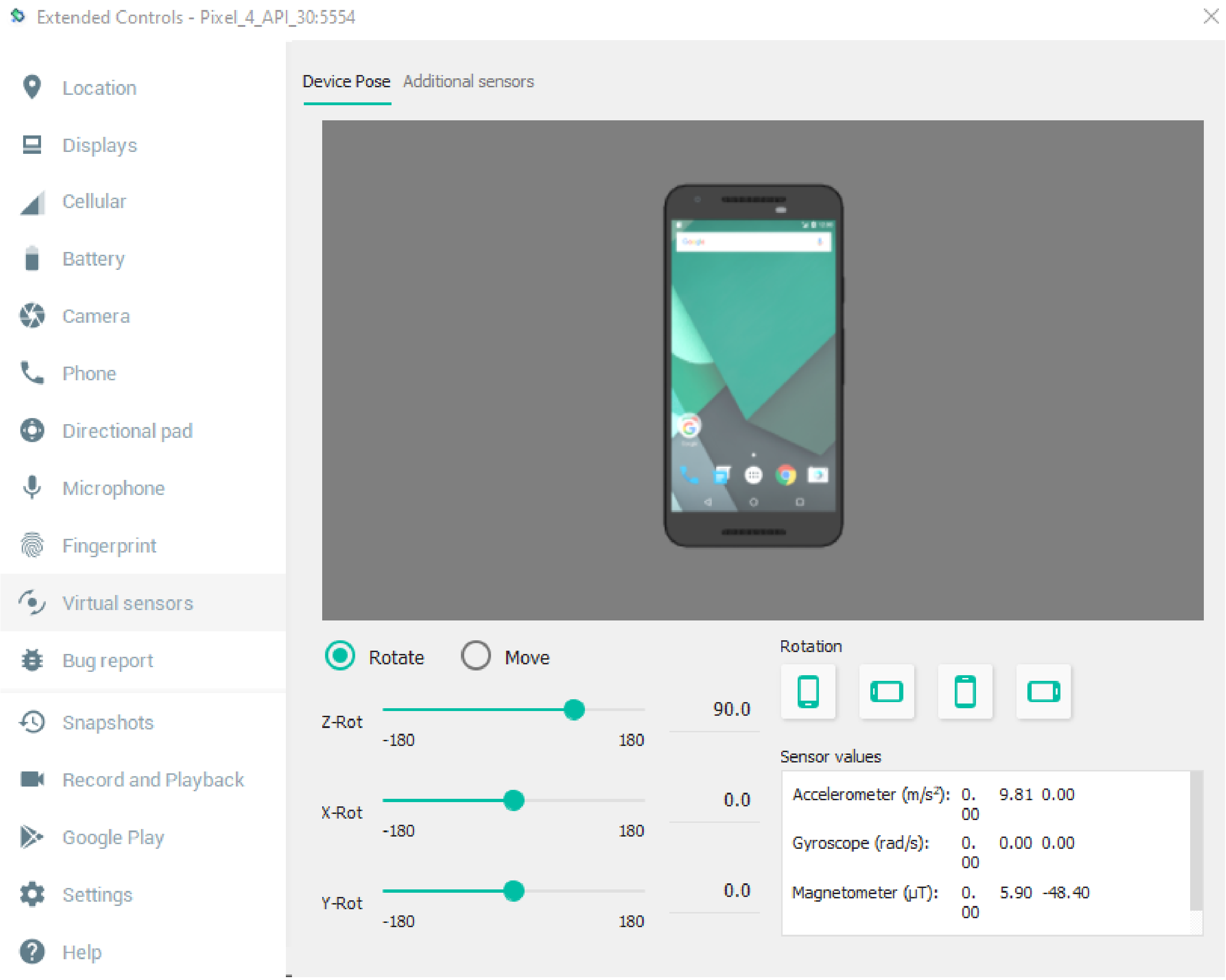
Accelerometer and Gyroscope


GYROSCOPE SENSING
ANGULAR ORIENTATION
$\overrightarrow{\Omega_Z}$
$\overrightarrow{\Omega_Y}$
$\overrightarrow{\Omega_X}$
ACCELEROMETER SENSING
AXIS ORIENTATION
$\overrightarrow{\text{ACC-Z}}$
$\overrightarrow{\text{ACC-Y}}$
$\overrightarrow{\text{ACC-X}}$

Accelerometer

- Accelerometer describe the velocity of the device, including the effects of gravity. Put simply, you can use accelerometer readings to tell if the device is moving in a particular direction.
- Mathematically, acceleration is a measurement of the change in velocity or speed divided by time.
- If you play a game, then you cannot have a good experience with a horizontal view. A landscape view provides users with more space to play a game on touch-enabled devices.
- While using a banking app, then portrait view is highly preferred by users compared to vertical as it is quite easy to add and read the information.

Example



● Rotate ○ Move

Z-Rot 90.0

X-Rot 0.0

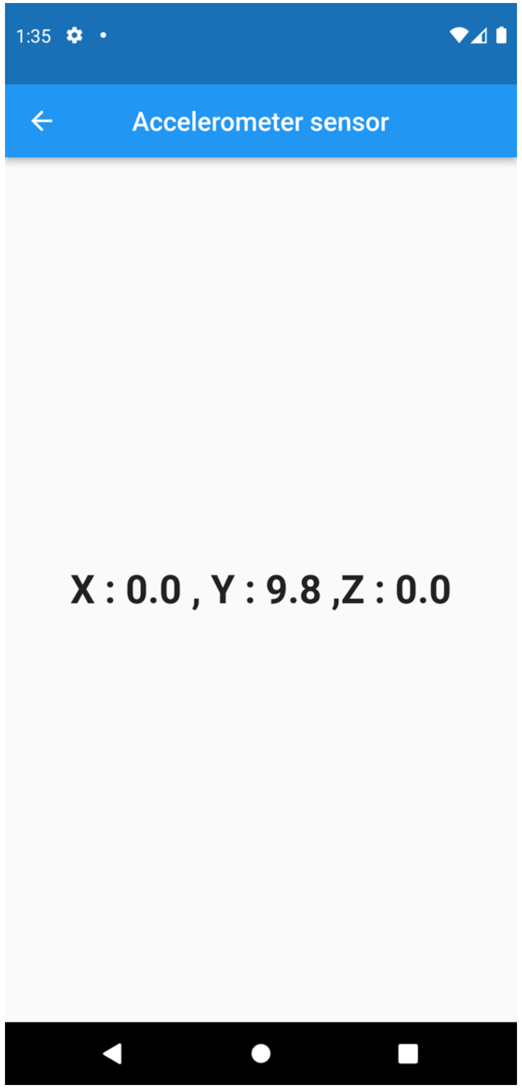
Y-Rot 0.0

Sensor values

Accelerometer (m/s²): 0.00 9.81 0.00

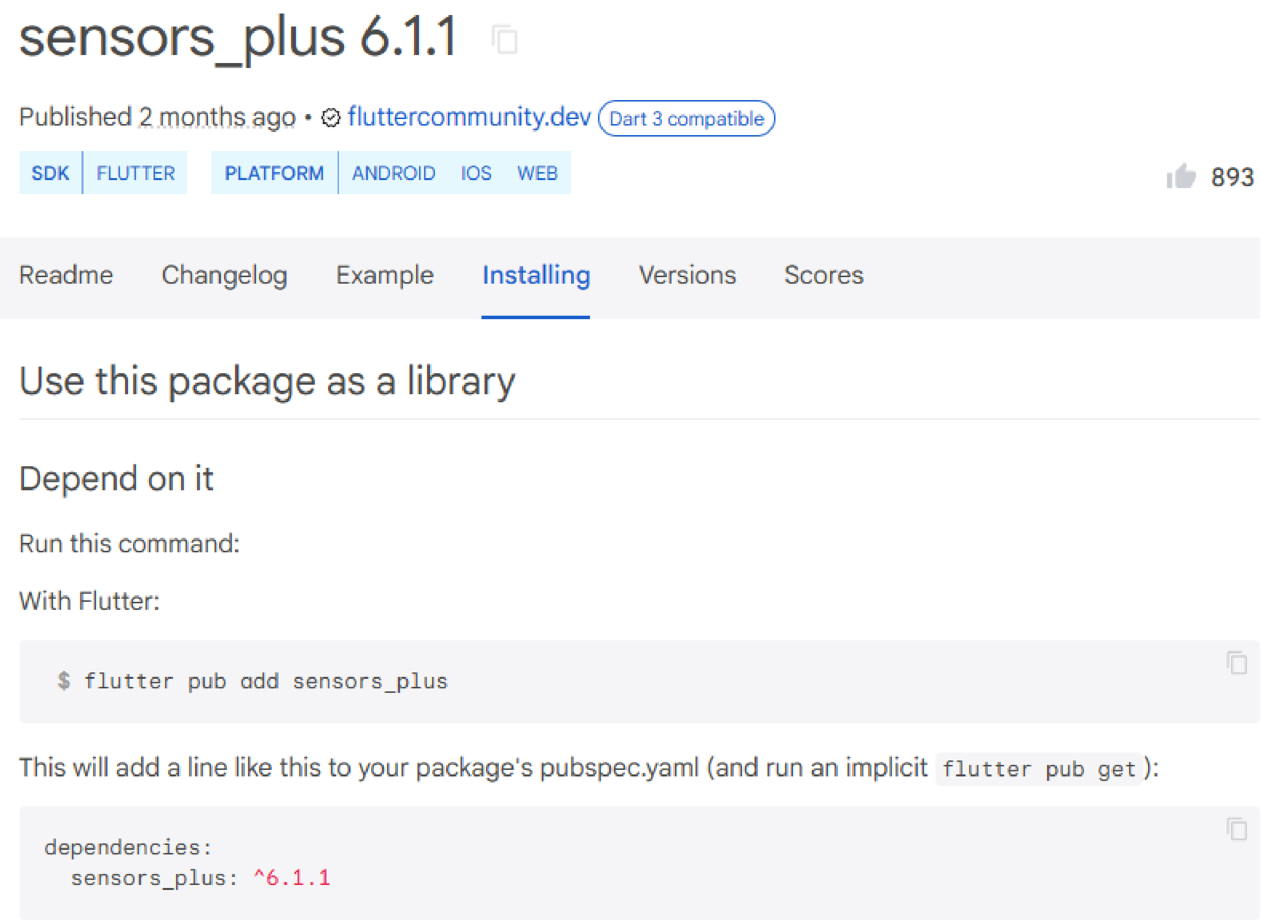
Gyroscope (rad/s): 0.00 0.00 0.00

Magnetometer (μT): 0.00 5.90 -48.40



X : 0.0 , Y : 9.8 ,Z : 0.0

Add dependency



sensors_plus 6.1.1

Published 2 months ago • fluttercommunity.dev Dart 3 compatible

Use this package as a library

Depend on it

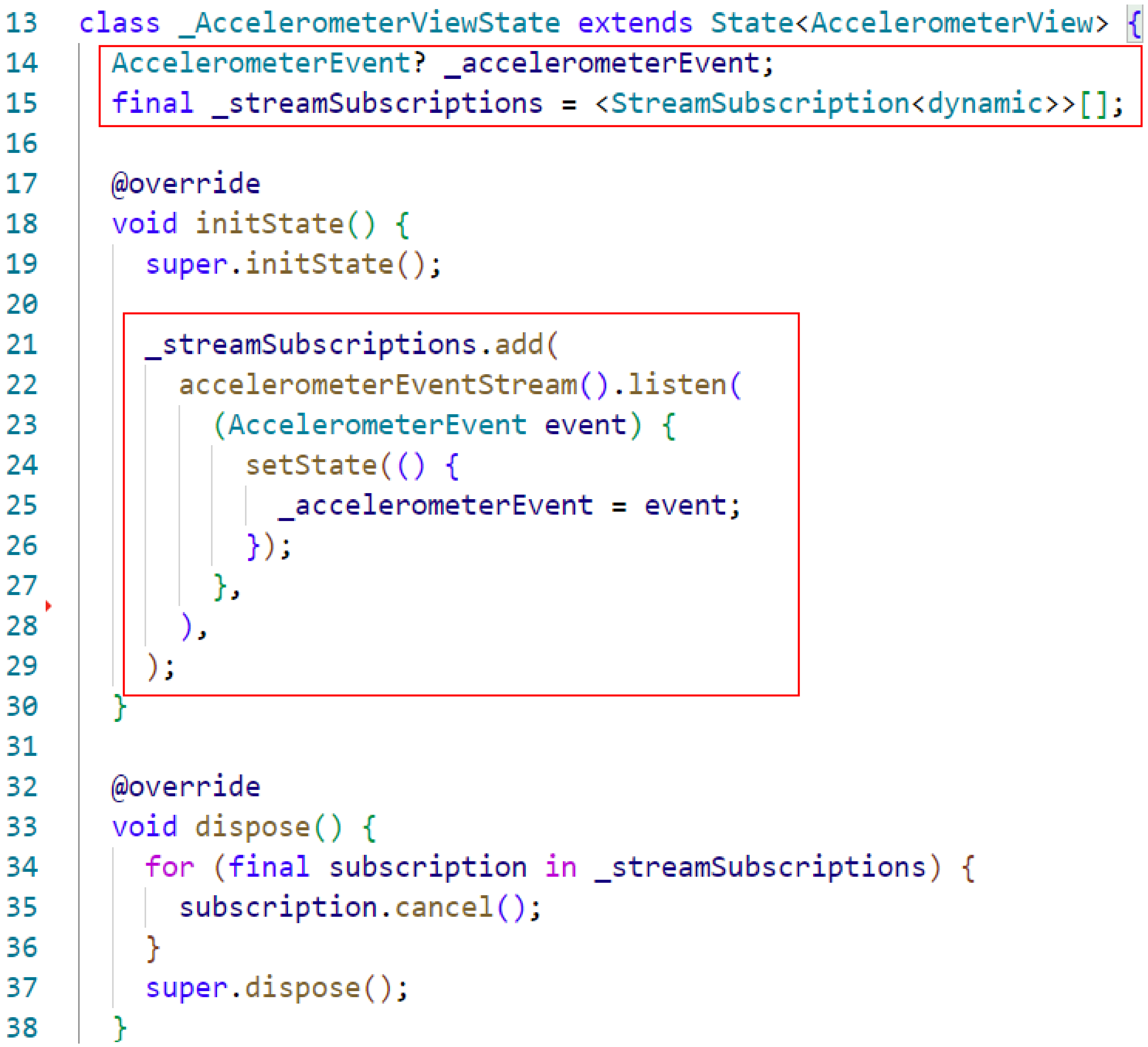
Run this command:

With Flutter:

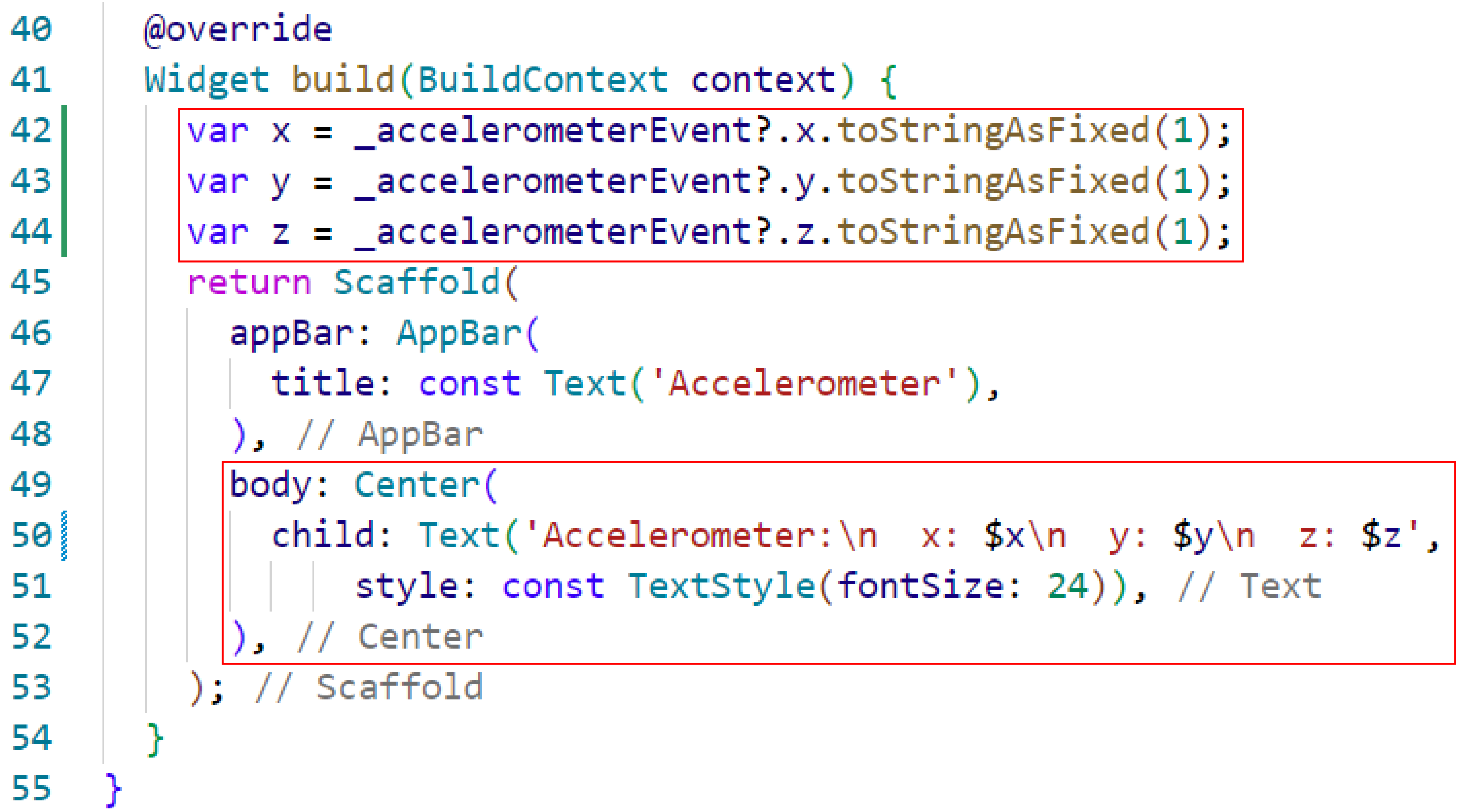
```
$ flutter pub add sensors_plus
```

This will add a line like this to your package's pubspec.yaml (and run an implicit `flutter pub get`):

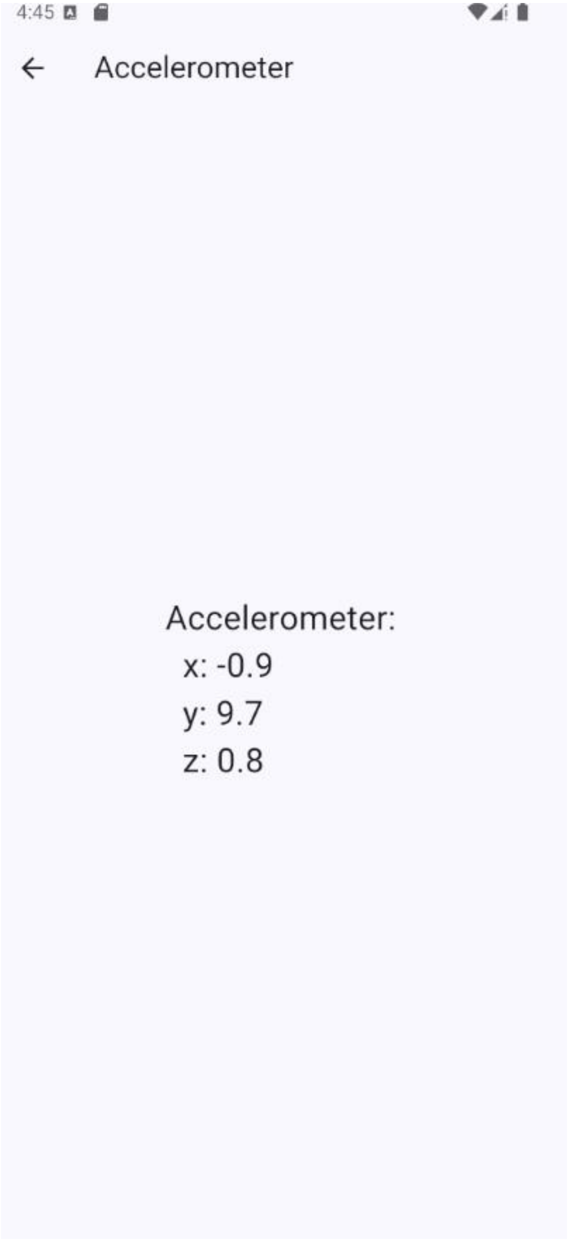
```
dependencies:
  sensors_plus: ^6.1.1
```



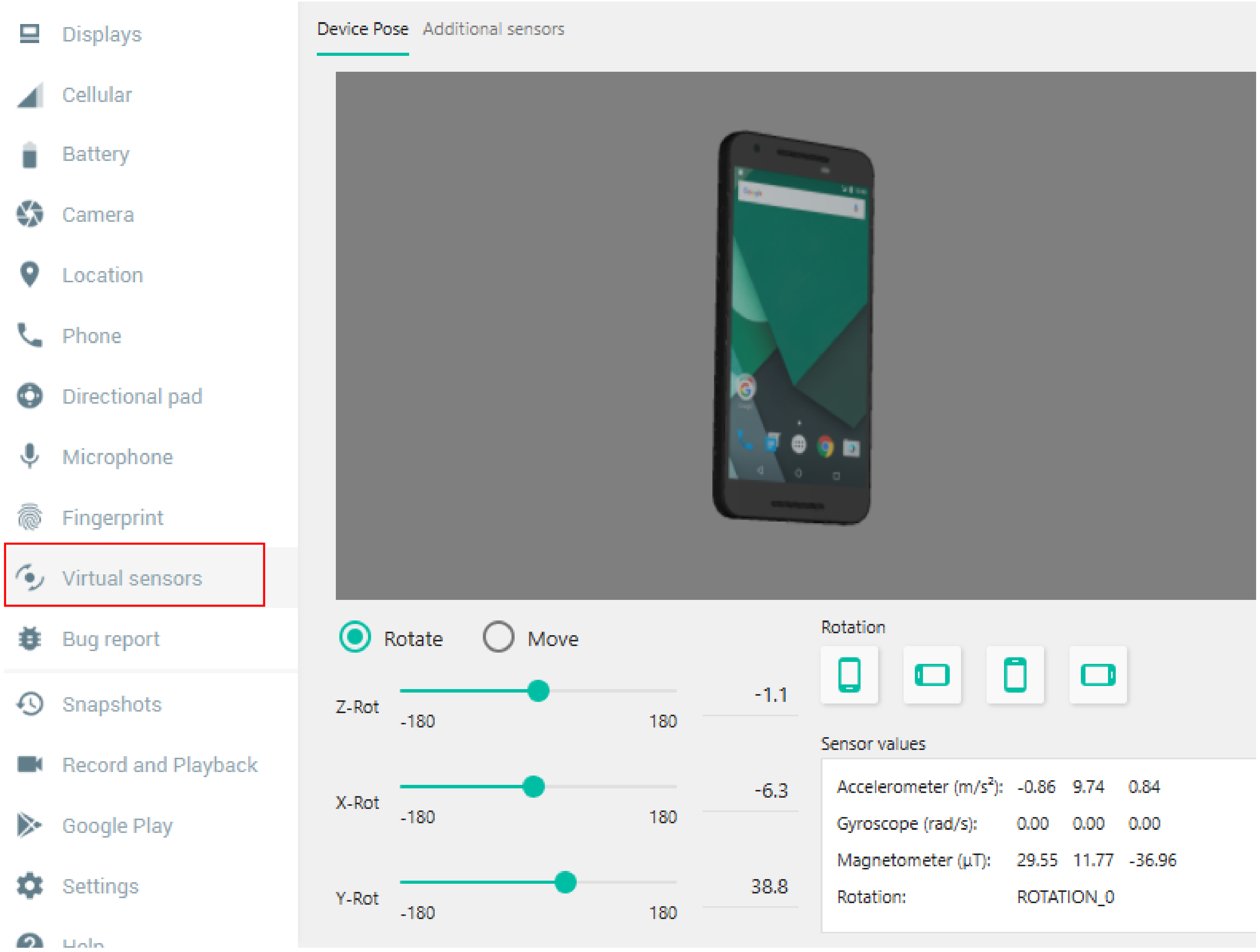
```
class _AccelerometerViewState extends State<AccelerometerView> {
  AccelerometerEvent? _accelerometerEvent;
  final _streamSubscriptions = <StreamSubscription<dynamic>>[];

  @override
  void initState() {
    super.initState();

    _streamSubscriptions.add(
      accelerometerEventStream().listen(
        (AccelerometerEvent event) {
          setState(() {
            _accelerometerEvent = event;
          });
        },
      ),
    );
  }

  @override
  void dispose() {
    for (final subscription in _streamSubscriptions) {
      subscription.cancel();
    }
    super.dispose();
  }
```



```dart
  @override
  Widget build(BuildContext context) {
    var x = _accelerometerEvent?.x.toStringAsFixed(1);
    var y = _accelerometerEvent?.y.toStringAsFixed(1);
    var z = _accelerometerEvent?.z.toStringAsFixed(1);
    return Scaffold(
      appBar: AppBar(
        title: const Text('Accelerometer'),
      ), // AppBar
      body: Center(
        child: Text('Accelerometer:\n  x: $x\n  y: $y\n  z: $z',
            style: const TextStyle(fontSize: 24)), // Text
      ), // Center
    ); // Scaffold
  }
}
```



Accelerometer:
x: -0.9
y: 9.7
z: 0.8



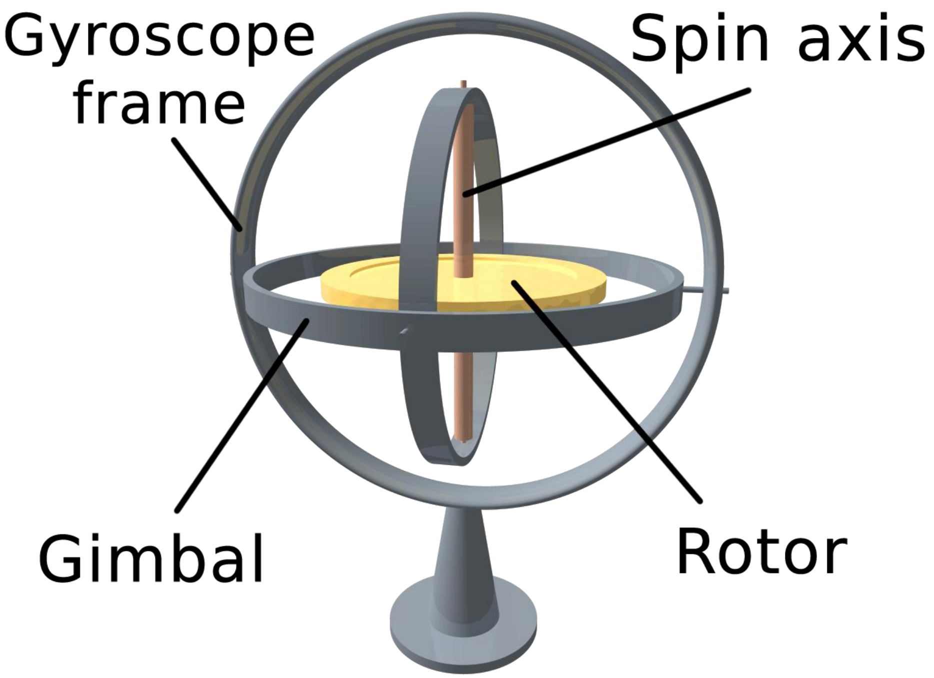
Z-Rot -1.1
X-Rot -6.3
Y-Rot 38.8

Sensor values

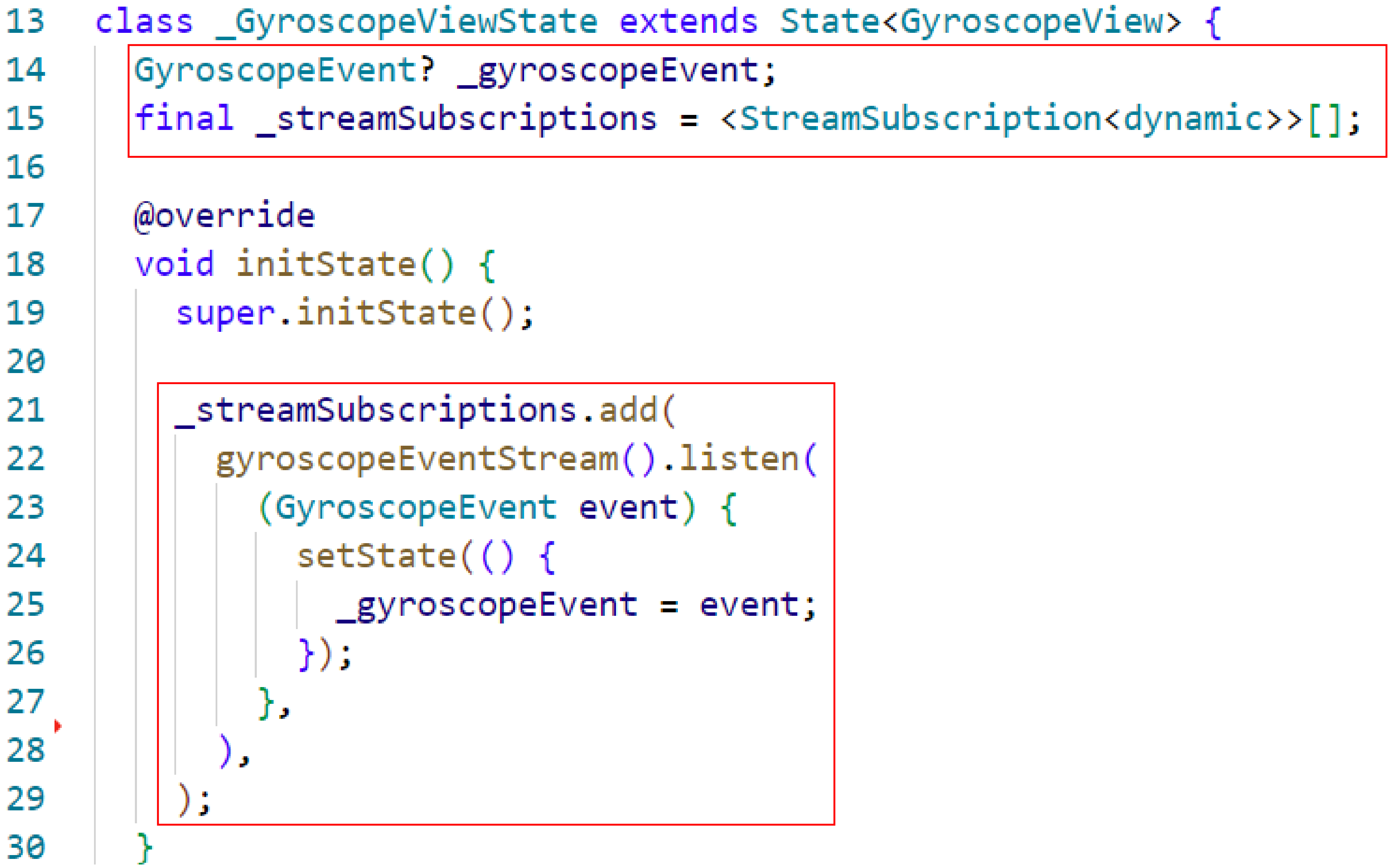
Accelerometer (m/s^2):	-0.86	9.74	0.84
Gyroscope (rad/s):	0.00	0.00	0.00
Magnetometer (µT):	29.55	11.77	-36.96
Rotation:	ROTATION_0		

Gyroscope

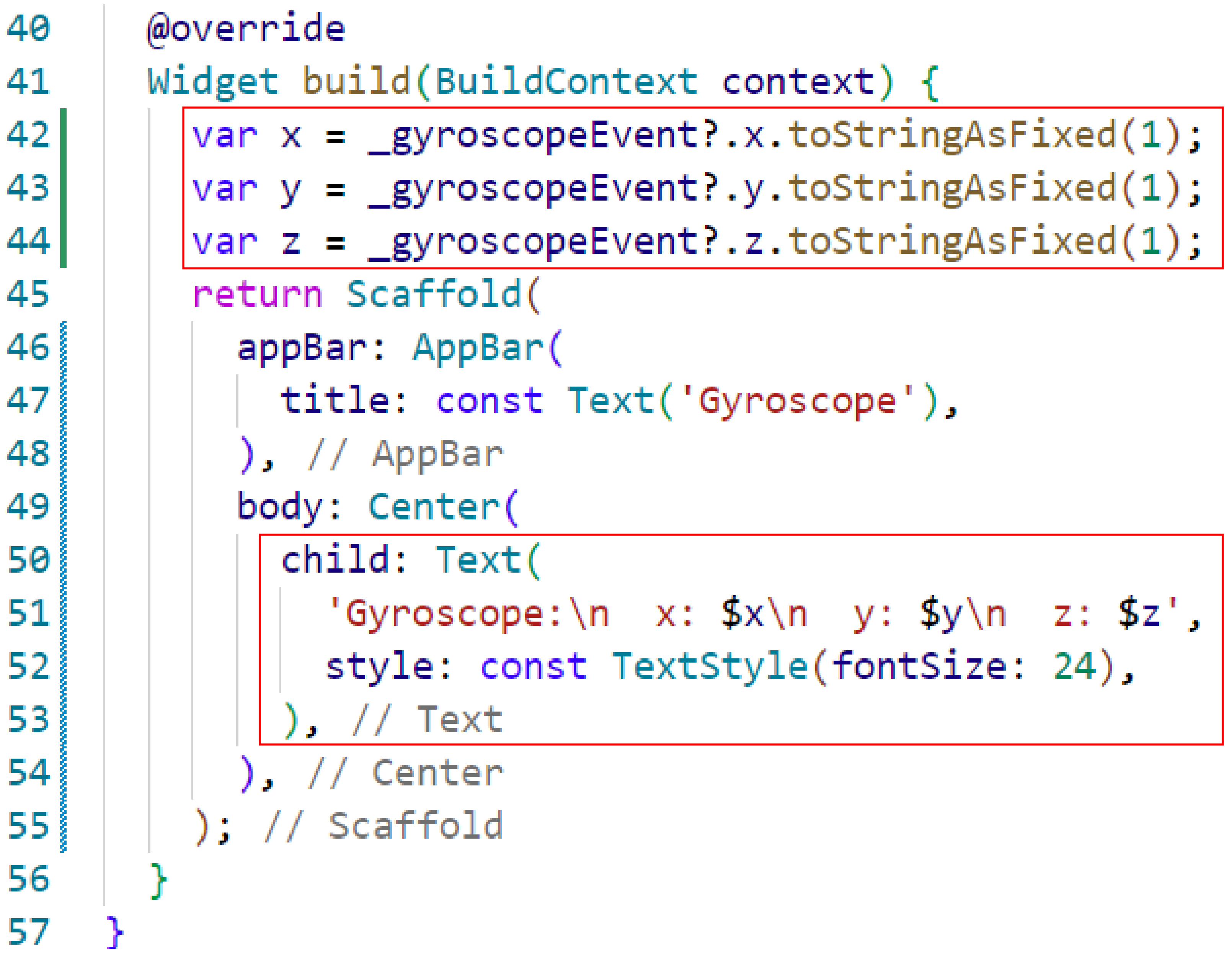
- Gyroscope describes the rotation of the device.



Code



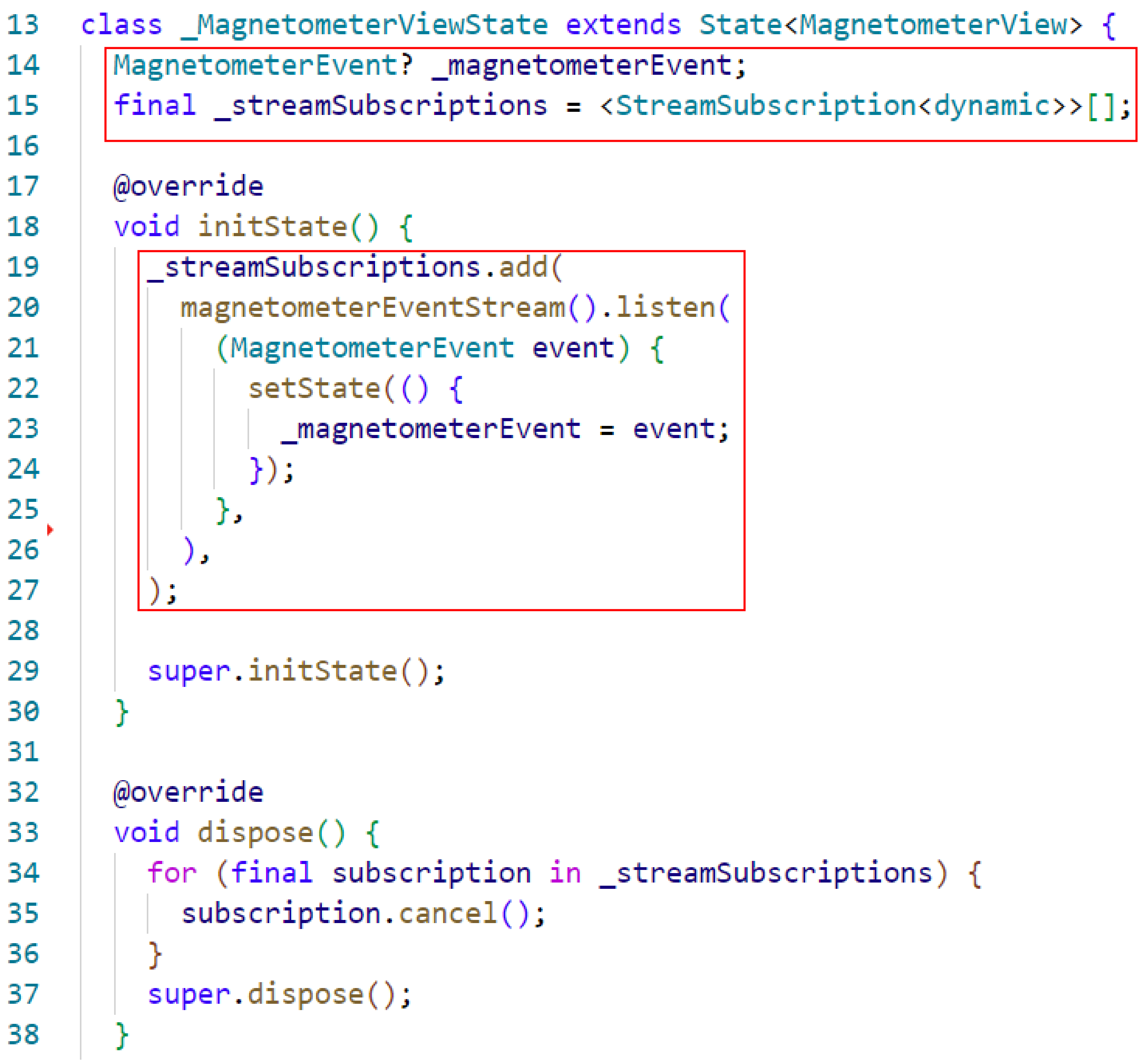
```
class _GyroscopeViewState extends State<GyroscopeView> {
  GyroscopeEvent? _gyroscopeEvent;
  final _streamSubscriptions = <StreamSubscription<dynamic>>[];

  @override
  void initState() {
    super.initState();

    _streamSubscriptions.add(
      gyroscopeEventStream().listen(
        (GyroscopeEvent event) {
          setState(() {
            _gyroscopeEvent = event;
          });
        },
      ),
    );
  }
```



```
  @override
  Widget build(BuildContext context) {
    var x = _gyroscopeEvent?.x.toStringAsFixed(1);
    var y = _gyroscopeEvent?.y.toStringAsFixed(1);
    var z = _gyroscopeEvent?.z.toStringAsFixed(1);
    return Scaffold(
      appBar: AppBar(
        title: const Text('Gyroscope'),
      ), // AppBar
      body: Center(
        child: Text(
          'Gyroscope:\n  x: $x\n  y: $y\n  z: $z',
          style: const TextStyle(fontSize: 24),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

Magnetometer

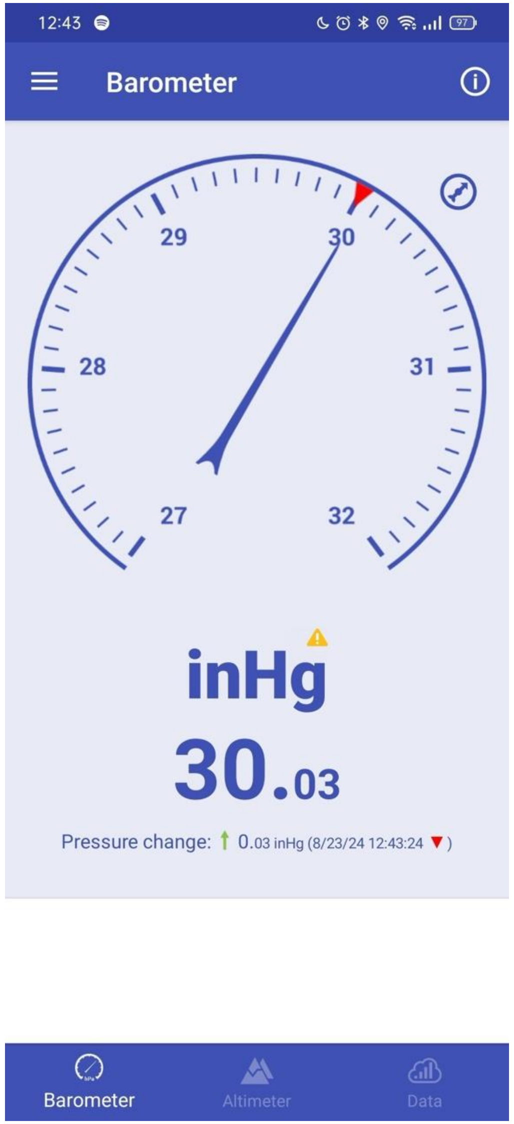
- A magnetometer sensor in an Android phone is a built-in component that measures the strength of the Earth's magnetic field along three axes (X, Y, and Z),
- A compass is one such device, one that measures the direction of an ambient magnetic field, in this case, the Earth's magnetic field.
- Other magnetometers measure the magnetic dipole moment of a magnetic material such as a ferromagnet, for example by recording the effect of this magnetic dipole on the induced current in a coil.



```dart
class _MagnetometerViewState extends State<MagnetometerView> {
  MagnetometerEvent? _magnetometerEvent;
  final _streamSubscriptions = <StreamSubscription<dynamic>>[];

  @override
  void initState() {
    _streamSubscriptions.add(
      magnetometerEventStream().listen(
        (MagnetometerEvent event) {
          setState(() {
            _magnetometerEvent = event;
          });
        },
      ),
    );

    super.initState();
  }

  @override
  void dispose() {
    for (final subscription in _streamSubscriptions) {
      subscription.cancel();
    }
    super.dispose();
  }
```

Barometer

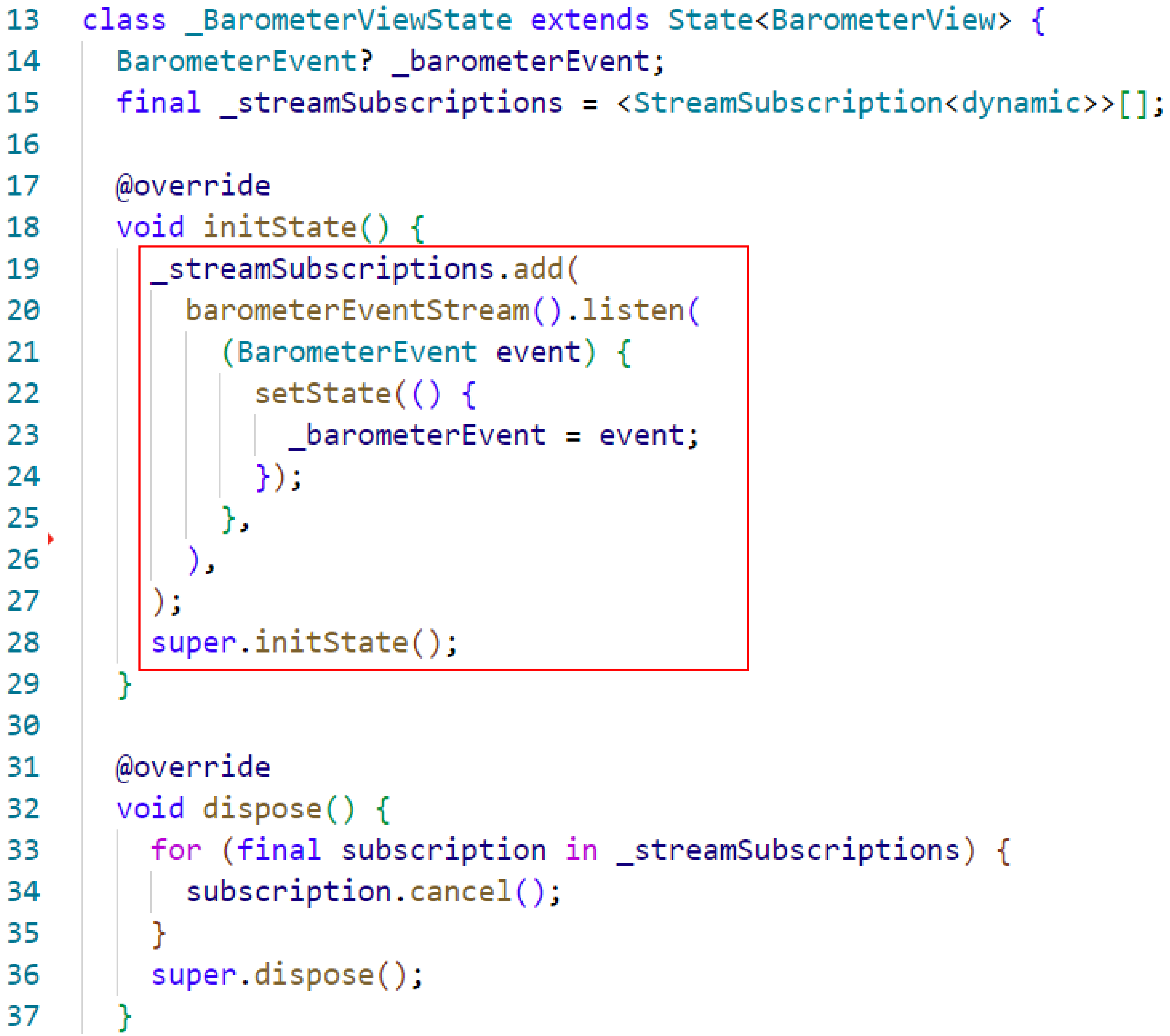
- A "barometer sensor" in an Android phone is a built-in sensor that measures atmospheric pressure, allowing the device to determine altitude and sometimes even provide basic weather predictions based on pressure changes; most modern smartphones include this sensor, making it accessible through various barometer apps on the Android platform.



inHg

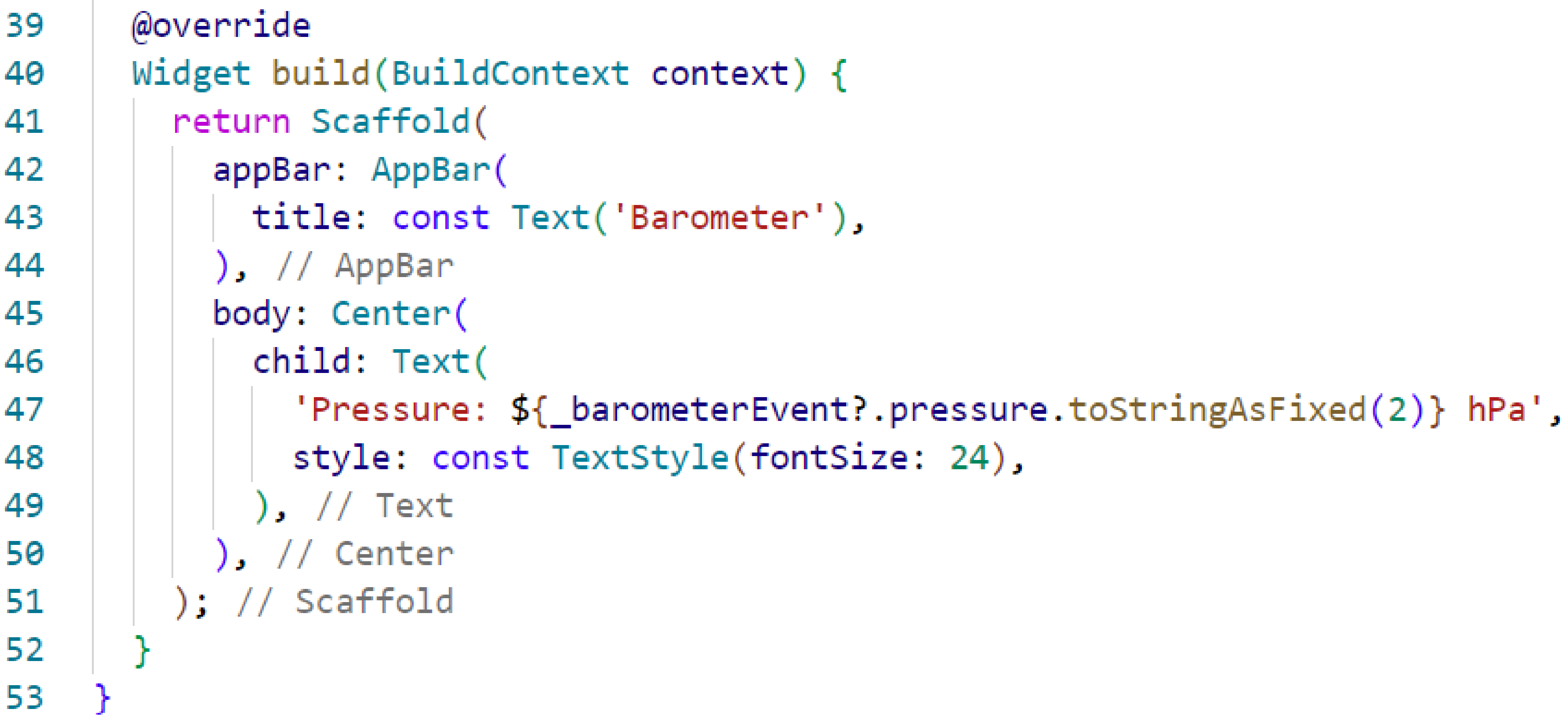
30.03

Pressure change: ↑ 0.03 inHg (8/23/24 12:43:24 ▼)

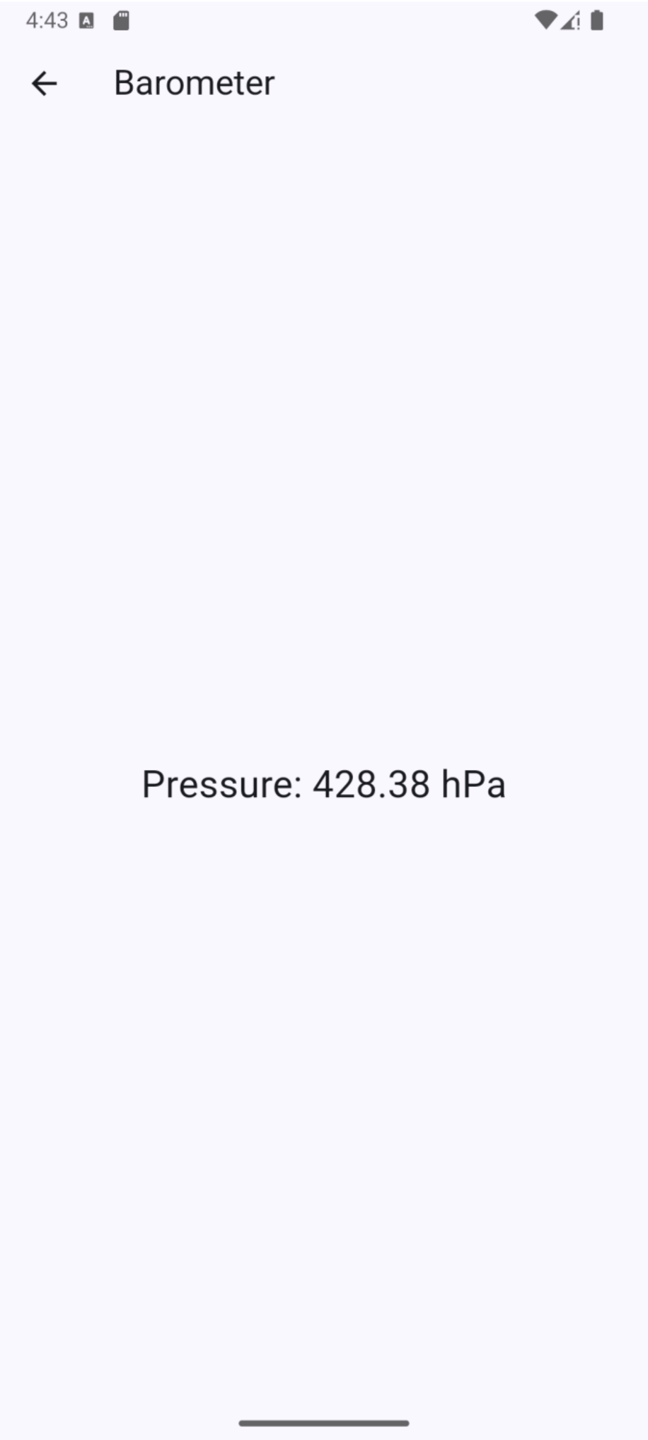


```dart
class _BarometerViewState extends State<BarometerView> {
  BarometerEvent? _barometerEvent;
  final _streamSubscriptions = <StreamSubscription<dynamic>>[];

  @override
  void initState() {
    _streamSubscriptions.add(
      barometerEventStream().listen(
        (BarometerEvent event) {
          setState(() {
            _barometerEvent = event;
          });
        },
      ),
    );
    super.initState();
  }

  @override
  void dispose() {
    for (final subscription in _streamSubscriptions) {
      subscription.cancel();
    }
    super.dispose();
  }
```

UI

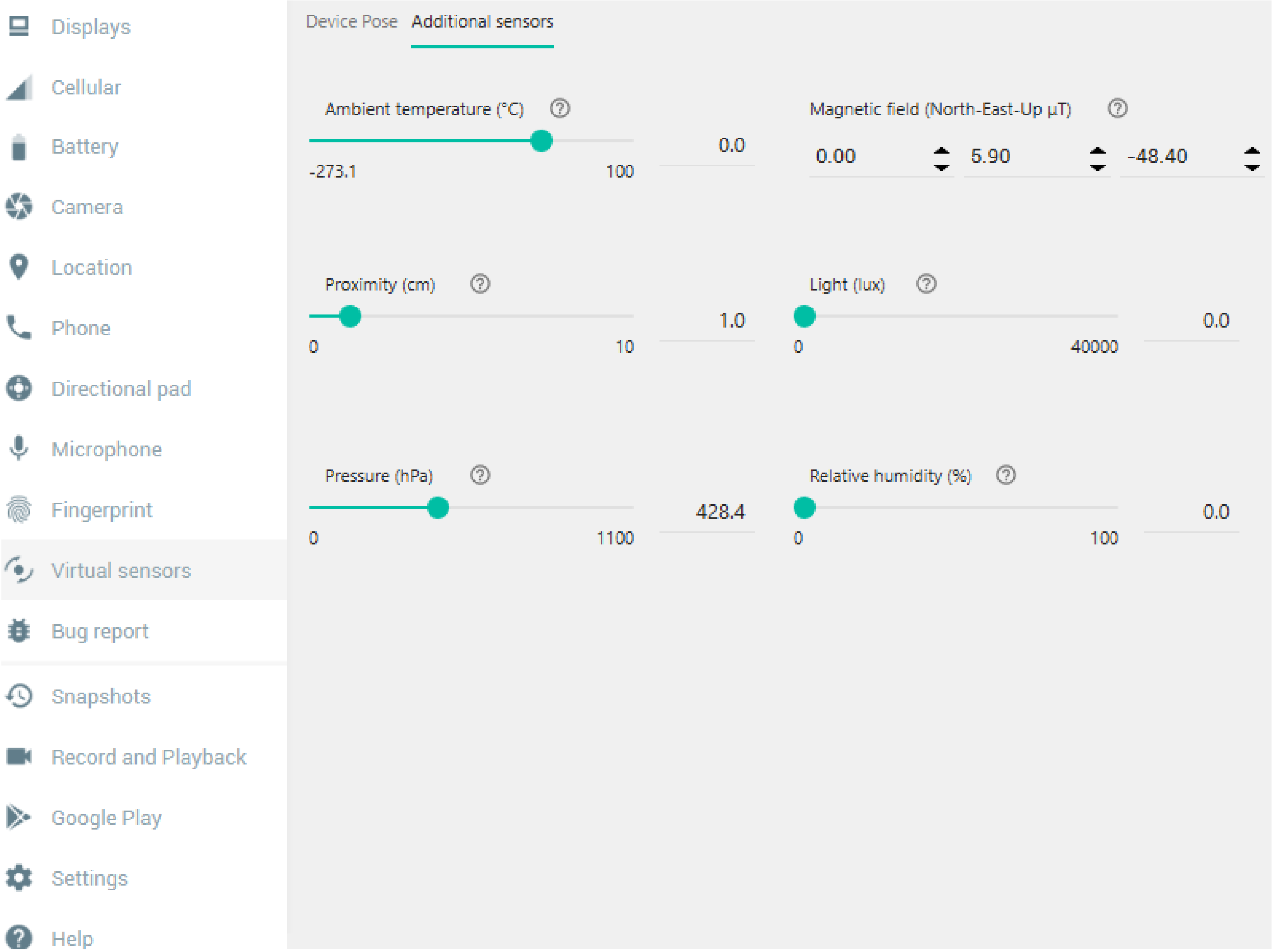


```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Barometer'),
      ), // AppBar
      body: Center(
        child: Text(
          'Pressure: ${_barometerEvent?.pressure.toStringAsFixed(2)} hPa',
          style: const TextStyle(fontSize: 24),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```



Barometer

Pressure: 428.38 hPa



Ambient temperature (°C) -273.1 100 0.0

Magnetic field (North-East-Up µT) 0.00 5.90 -48.40

Proximity (cm) 0 10 1.0

Light (lux) 0 40000 0.0

Pressure (hPa) 0 1100 428.4

Relative humidity (%) 0 100 0.0