



Capteurs - Angle

[Mise à jour le 30/6/2022]

1. Généralités

En géométrie, la notion générale d'angle se décline en plusieurs concepts. Dans son sens ancien, l'angle est une figure plane, portion de plan délimitée par deux demi-droites. C'est ainsi que l'on parle des angles d'un polygone. Cependant, l'usage est maintenant d'employer le terme « **secteur angulaire** » pour une telle figure. L'angle peut désigner également une portion de l'espace délimitée par deux plans (angle dièdre). La mesure de tels angles porte couramment, mais abusivement le nom d'angle, elle aussi. [Wikipédia](#)



Un capteur angulaire est un capteur servant à recueillir une image de l'angle mesuré. L'information angulaire peut être sous forme numérique ou analogique. [Composant électronique](#)



- Ressource à consulter sur **Wikiversité** : [capteur de position](#)

2. Capteurs analogiques

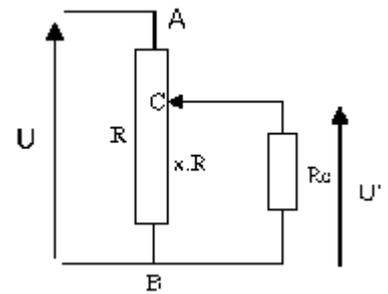
2.1 Potentiomètre



- **Source** : [wiki](#)

Module capteur rotatif Gravity DFRobot délivrant une valeur analogique en fonction de la position du potentiomètre. Il est livré avec un cordon 3 contacts pour le raccordement.

- **Distributeur** : [Gotronic](#)



- **Caractéristiques**

- Alimentation: 3,3 à 5 Vcc
- Rotation: 300°
- Dimensions: 28 x 22 x 23 mm

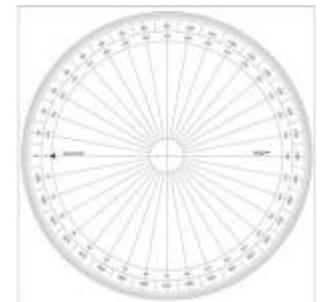
- **Modèle**

- $U' = x * U$ si ($R_c \gg R$)



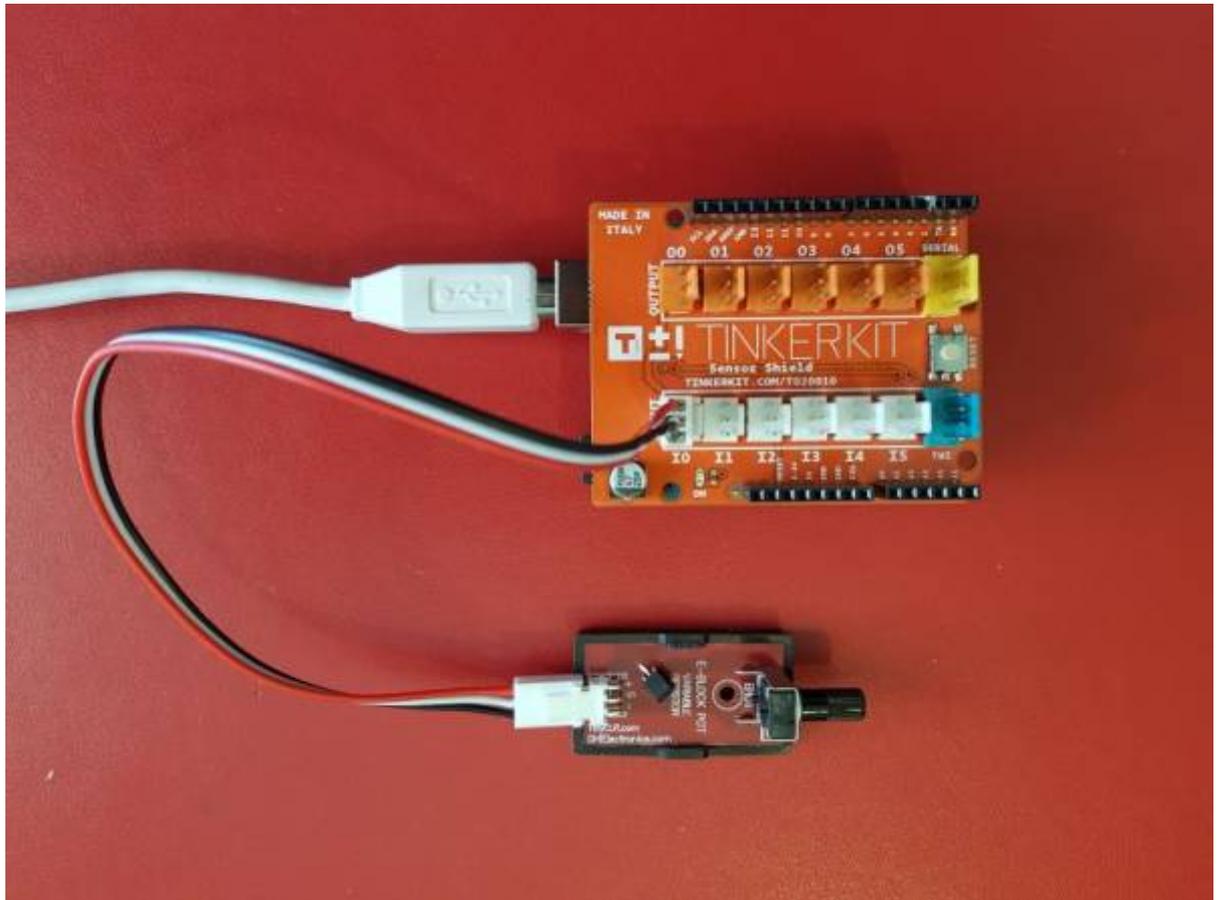
- **Aide pour la *simulation de la chaîne de mesure***

- Les équations de la chaîne de mesure sont téléchargeables [A venir](#)
- Le modèle à simuler est téléchargeable [A venir](#)



- **Programmation d'une carte Arduino Uno R3**

- Bibliothèques à installer dans l'IDE : aucune
- Connexion à un shield [Tinkerkit v2](#)



◦ Un premier exemple



pot.cpp

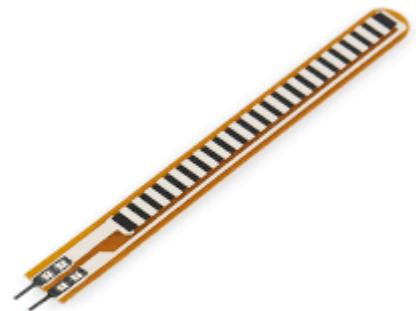
```
/*  
Titre : Montage potentiométrique  
Date : 18/6/2021  
Objectif : afficher la position angulaire d'un potentiomètre  
Matériels : capteur rotatif Gravity DFRobot, carte Arduino Uno, etc.  
PlatformIO  
- Répertoire : potentiometre  
- Fichier : main.cpp  
*/  
void setup()  
{  
  Serial.begin(9600); // Débit binaire : 9600 bps  
}  
  
void loop()  
{  
  uint16_t N;  
  double angle;  
  // Lecture  
  N=analogRead(A0); //LM35 connecté à Analog 0  
  // Traitement  
  // Sensibilité du capteur S = 5/300  
  angle = (double) N * (3/(10.24));  
}
```

```
// Ecriture
Serial.print("Angle:"); // Affiche l'angle sur le moniteur
Serial.print(angle);
Serial.println("°");
delay(1000);
}
```



 [Télécharger](#) le projet PlatformIO pour VSCode.

2.2 Flex sensor



- Source : [wiki](#)

 Note

- Distributeur : [????](#)
- Caractéristiques
 - Alimentation :



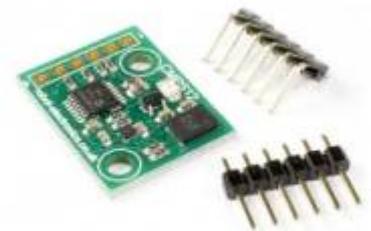
- Documentation
 - PDF à télécharger [ici](#)
- Bibliothèques à télécharger dans l'IDE [????](#)



- **Un premier exemple pour tester le capteur avec l'IDE Arduino**
→ Fichier → Exemples →
- **Mise en oeuvre du capteur avec un afficheur OLED**
 - *Description* :
 - *Matériels*
 - Carte à microcontrôleur : [Adafruit Feather Huzzah ESP8266 + Support Particle](#)
 - Afficheur : [Adafruit OLED SH1107](#)
 - *Code Arduino*

*.cpp

3. Capteurs numériques (Gyroscope)



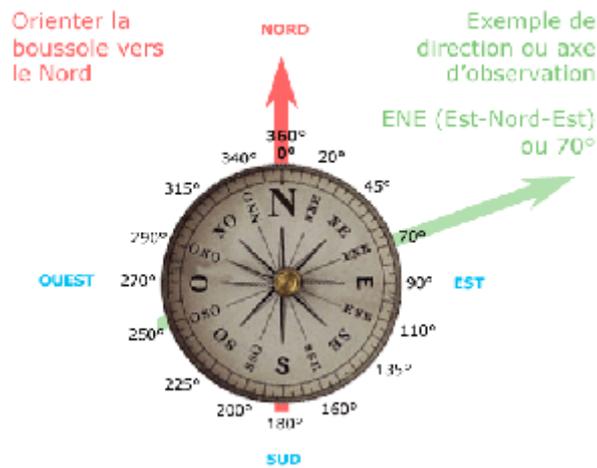
3.1 CMPS12

- **Source** : [Robot Electronics](#)

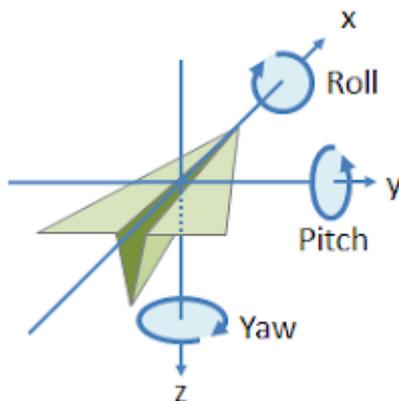
Module CMPS12 de 4ème génération basé sur un circuit Bosch BNO055 intégrant une boussole 3 axes, un **gyroscope 3 axes** et un accéléromètre 3 axes. La direction par rapport au nord est représentée par un nombre unique, soit en I²C ou par port série.

Le module délivre les informations suivantes :

- **La direction par rapport au nord** :
 - sur 8 bits : 0 à 255
 - sur 16bits : 0 à 3599 (représente 0 à 359,9°)

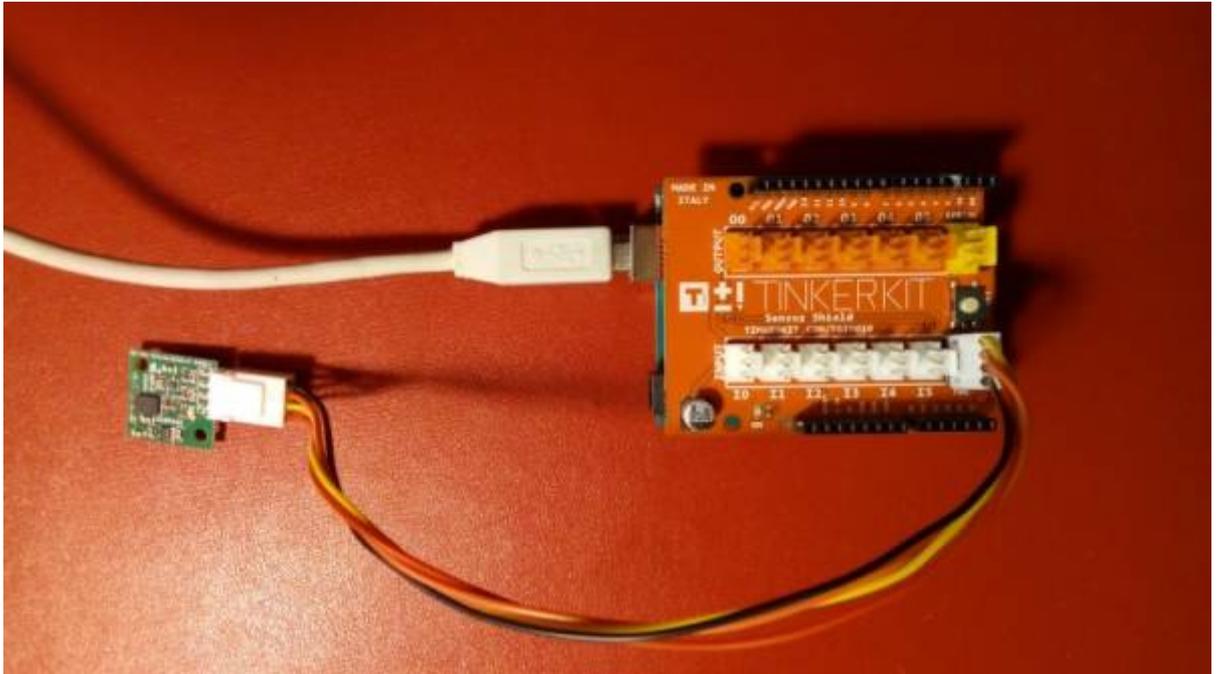


- **Pitch** : +/- 0 à 90° ou +/- 0 à 180°
- **Roll** : +/- 0 à 90°



Axis of Rotation	Euler Angle Name	Euler Angle Symbol
x	Roll	u
y	Pitch	v
z	Yaw	w

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
 - Alimentation: 3,3 ou 5 Vcc
 - Consommation: 18 mA
 - Résolution: 0,1°
 - Précision: 1 %
 - Interface
 - I²C (jusqu'à 400 kHz)
 - série (9600, 19200 et 38400 bps)
 - Dimensions: 25 x 18 x 3 mm
- **Documentation** : [pdf à télécharger](#)
- **Programmation d'une carte Arduino Uno R3**
 - Bibliothèques à installer dans l'IDE : aucune
 - Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno.



◦ Un premier exemple



*.cpp

```

/*****
*   CMPS12 I2C example for Arduino   *
*   By James Henderson, 2014       *
*****/

#include <Wire.h>

#define CMPS12_ADDRESS 0x60 // Adresse du CMPS12 sur 7 bits
#define ANGLE_8 1 // Registre contenant la valeur de l'angle
sur 8bits // Sert de référence pour la lecture dans
la table

unsigned char high_byte, low_byte, angle8;
char pitch, roll;
unsigned int angle16;

void setup()
{
  Serial.begin(9600); // Port série
  Wire.begin();
}

void loop()
{
  Wire.beginTransmission(CMPS12_ADDRESS); // Démarrage de la
communication avec le CMPS12
  Wire.write(ANGLE_8); // Envoi du numéro du

```

```
premier registre de la // table à partir duquel on
souhaite lire
Wire.endTransmission();

// On demande la lecture de 5 octet à partir du registre 0x01
// afin d'obtenir l'angle sur 8bits,
// l'angle sur 16 bits, le pitch et le roll
Wire.requestFrom(CMPS12_ADDRESS, 5);

while(Wire.available() < 5); // On attend la fin de la
transmission des 5 octets

angle8 = Wire.read(); // Lecture des octets
high_byte = Wire.read();
low_byte = Wire.read();
pitch = Wire.read();
roll = Wire.read();

angle16 = high_byte; // Calcul de l'angle sur 16bits
angle16 <<= 8;
angle16 += low_byte;

Serial.print("roll: "); // Affichage de roll
Serial.print(roll, DEC);

Serial.print(" pitch: "); // Affichage de pitch
Serial.print(pitch, DEC);

Serial.print(" angle full: "); // Affichage de l'angle codé
sur 16 bits avec décimale
Serial.print(angle16 / 10, DEC);
Serial.print(".");
Serial.print(angle16 % 10, DEC);

Serial.print(" angle 8: "); // Affichage de l'angle codé
sur 8 bits
Serial.println(angle8, DEC);

delay(1000); // Attente entre deux mesures
}
```



Télécharger le projet PlatformIO pour VSCode.

- **Programmation d'une carte FEZ Duino sous TinyCLR OS avec l'IDE Visual Studio Community**

A venir

3.2 Module L3GD20H



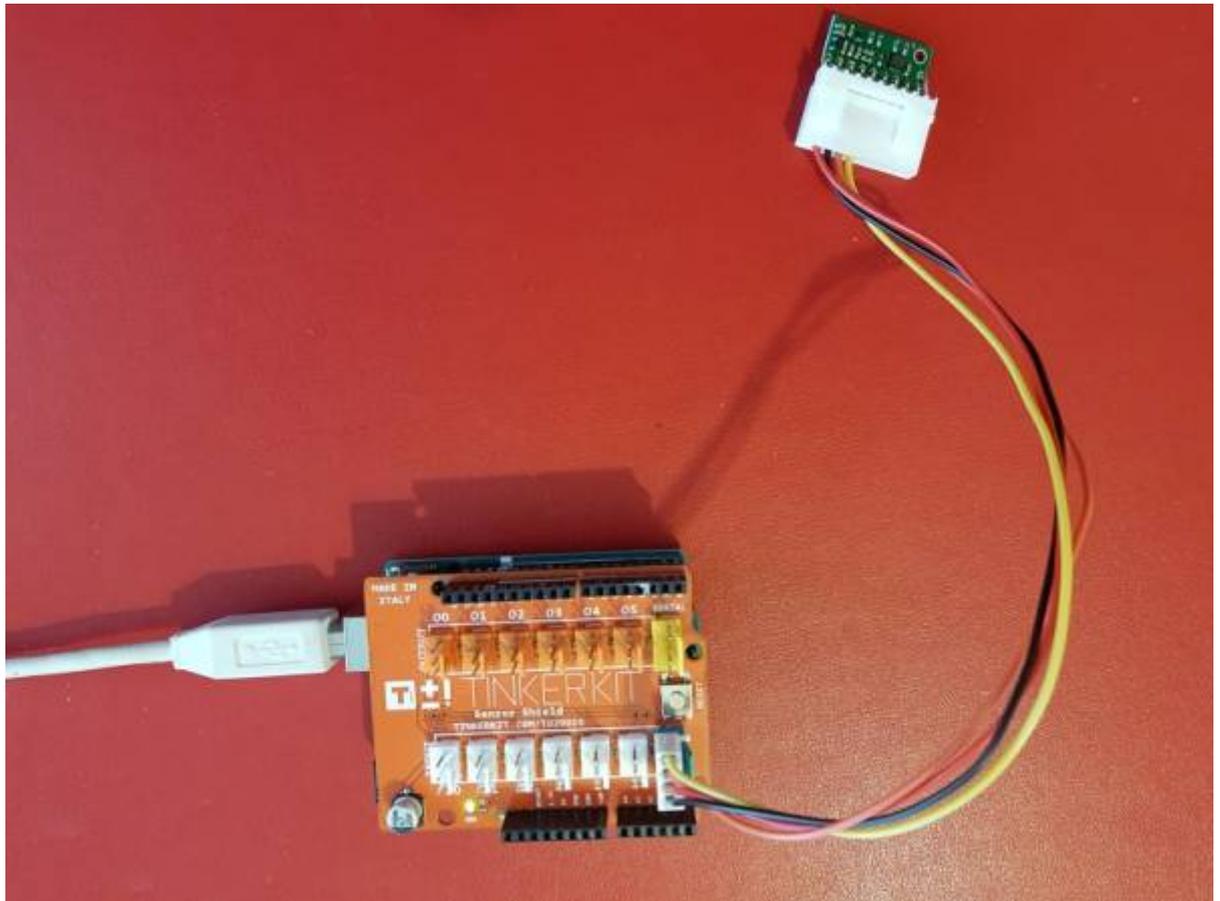
- **Source** : [Wiki](#)

Ce module gyroscope 3 axes de Pololu est basé sur le circuit L3GD20H et communique avec un microcontrôleur via un port I2C.

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
 - Alimentation: 2,5 à 5,5 Vcc
 - Consommation: environ 6 mA
 - Interface: I2C
 - Sortie I2C: lecture sur 16 bits par axe x, y, z
 - Plages de mesure: ± 245 , ± 500 ou $\pm 2000^\circ/s$
 - Dimensions: 23 x 13 x 11 mm



- **Documentation**
 - Fichier Acrobat Reader à télécharger [ici](#)
- **Programmation d'une carte Arduino Uno R3**
 - *Bibliothèque à installer dans l'IDE*
L3G by Pololu Version 3.0.0 INSTALLED
Arduino library for Pololu L3G4200D, L3GD20, and L3G4200D boards This is a library for an Arduino-compatible controller that interfaces with L3GD20H, L3GD20, and L3G4200D gyros on Pololu board.
 - *Connexion à un shield Tinkerkit v2.*



◦ Un premier exemple 

Arduino Examples → Examples from Custom Libraries → L3G → **Serial.ino**

Exemple de résultat attendu

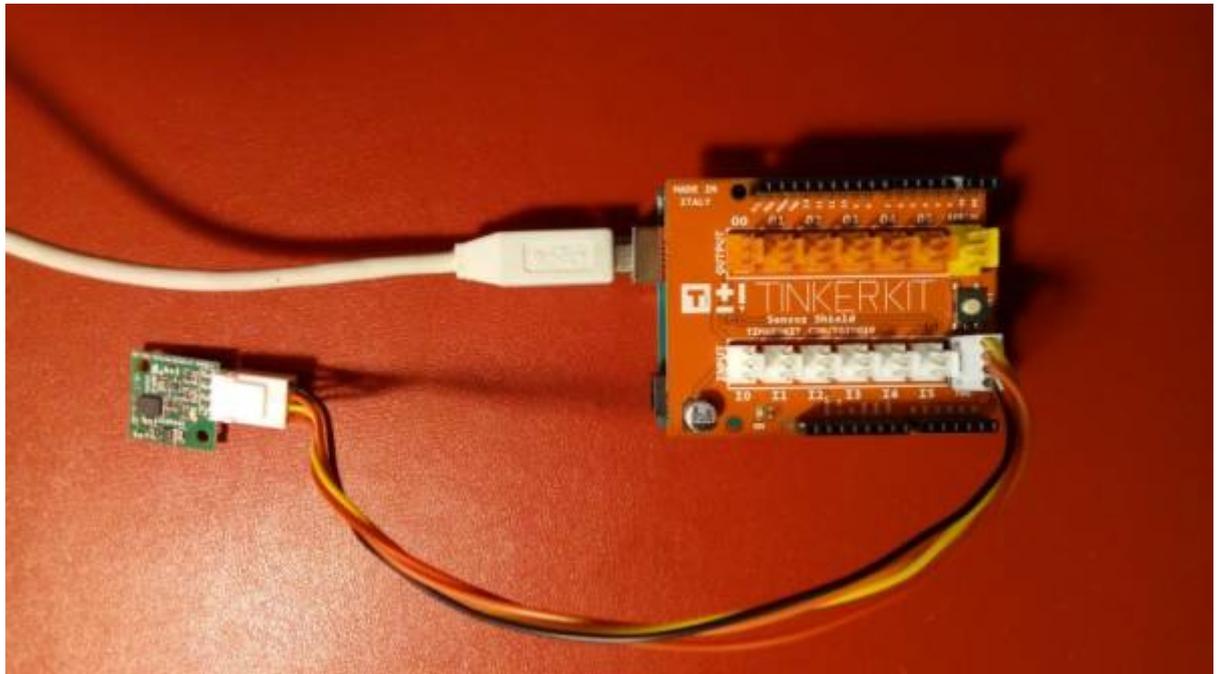
```
CONSOLE DE DÉBOGAGE  SORTIE  TERMINAL
G·X: 20·Y: -79·Z: 91
G·X: 12·Y: -48·Z: 83
G·X: 10·Y: -38·Z: 87
```

C#

- Programmation d'une carte **FEZ Duino** sous **TinyCLR OS** avec l'IDE Visual Studio Community

A venir

3.3 Module BNO055



◦ Un premier exemple (I2C) 

Arduino Examples → Examples from Custom Libraries → Adafruit_BNO055 → **sensorapi.ino**

Exemple de résultat attendu

```
X: 356.0000 Y: 10.1875 Z: 22.0000 Sys:3 G:3 A:0 M:2  
X: 2.8125 Y: -1.0625 Z: 29.1250 Sys:2 G:3 A:0 M:2  
X: 7.1875 Y: -6.0625 Z: 32.4375 Sys:2 G:3 A:0 M:2  
X: 5.6250 Y: -5.7500 Z: 34.3125 Sys:2 G:3 A:0 M:2  
X: 7.7500 Y: -7.4375 Z: 32.0625 Sys:2 G:3 A:0 M:2
```

C#

- **Programmation d'une carte FEZ Duino sous TinyCLR OS avec l'IDE Visual Studio Community**

A venir

From:
<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:
<http://webge.fr/dokuwiki/doku.php?id=materiels:capteurs:angle:angle&rev=1656585359>

Last update: **2022/06/30 12:35**

