

# Magnetómetro

Página extraída de Federico Coca [Guía de Trabajo de Microbit](#) CC-BY-SA

Este módulo permite acceder a la brújula electrónica incorporada. Antes de utilizarla, la brújula debe estar calibrada; de lo contrario, las lecturas pueden ser erróneas.

**Advertencia.** Calibrar la brújula hará que su programa se detenga hasta que se complete la calibración. La calibración consiste en un pequeño juego para dibujar un círculo en la pantalla LED girando el dispositivo.

Las funciones son:

- `microbit.compass.calibrate()`  
Inicia el proceso de calibración. El mensaje Tilt to Fill Screen (Inclinar para llenar la pantalla) se desplazará en la pantalla que el usuario debe rellenar completamente moviendo el dispositivo.
- `microbit.compass.is_calibrated()`  
Devuelve `True` si la brújula se ha calibrado correctamente, y devuelve `False` en caso contrario.
- `microbit.compass.clear_calibration()`  
Deshace la calibración, haciendo que la brújula vuelva a estar descalibrada.
- `microbit.compass.get_x()`
- `microbit.compass.get_y()`
- `microbit.compass.get_z()`  
Da la lectura, como un número entero positivo o negativo, de la intensidad del campo magnético en el eje especificado dependiendo de la dirección del campo. La medida se da en nano teslas.
- `microbit.compass.heading()`  
Da el rumbo de la brújula, calculado a partir de las lecturas anteriores, como un número entero en el rango de 0 a 360, representando el ángulo en grados, en el sentido de las agujas del reloj, con el norte como 0.
- `microbit.compass.get_field_strength()`  
Devuelve un valor entero de la magnitud del campo magnético alrededor del dispositivo expresada en nano tesla.

## Ejemplos de actividades con la brújula

Antes de nada debemos calibrar la brújula para después mostrar la posición de la micro:bit utilizando las flechas predefinidas.

## Brújula indicando el norte

```
from microbit import *

# Antes de nada calibrar
compass.calibrate()

# Mantener la aguja apuntando aproximadamente en la dirección correcta.
while True:
    sleep(100)
    aguja = ((15 - compass.heading()) // 30) % 12
    display.show(Image.ALL_CLOCKS[aguja])
```

## Brújula N,S,E,O

El programa, de muy poca precisión es el siguiente:

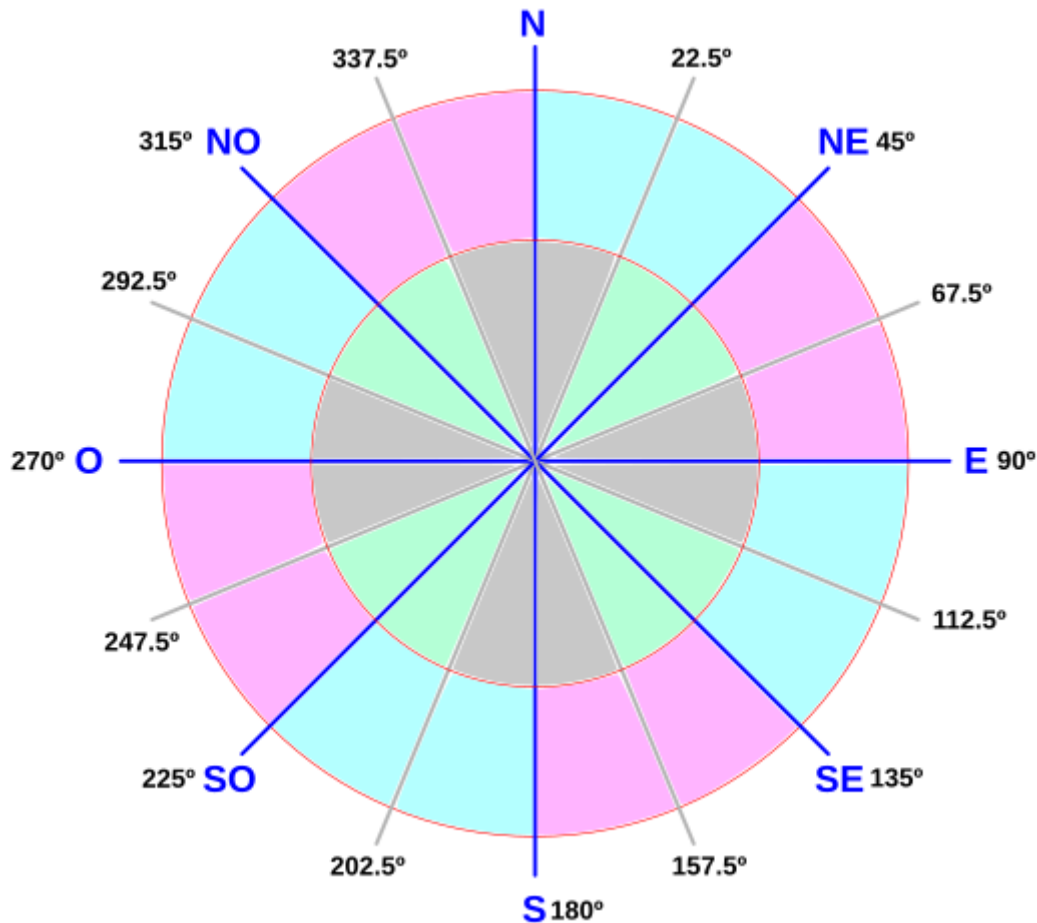
```
from microbit import *

compass.calibrate()
while True:
    angulo = compass.heading()
    if angulo < 45:
        display.show("N")
    elif angulo < 135:
        display.show("E")
    elif angulo < 225:
        display.show("S")
    elif angulo < 315:
        display.show("O")
    else:
        display.show("N")
```

## Brújula a 45º

Lo que vamos a hacer ahora es localizar las siguientes ocho direcciones, separadas entre si un ángulo de 45º: noroeste (NE), oeste (O), suroeste (SO), sur (S), sureste (SE), este (E), noreste (NE), norte (N). El margen o ángulo que va a englobar cada dirección estará ajustado dentro de otro ángulo de 45º dividido en dos iguales de 22.5º respecto a la dirección principal. Un gráfico nos

aclara mejor la idea



En el editor Mu también son válidas las definiciones predefinidas para las flechas.

El programa es:

```
from microbit import *

compass.calibrate()
while True:
    angulo = compass.heading()
    if angulo > 22.5 and angulo <= 67.5:
        display.show(Image.ARROW_NE)
    elif angulo > 67.5 and angulo <= 112.5:
        display.show(Image.ARROW_E)
    elif angulo > 112.5 and angulo <= 157.5:
        display.show(Image.ARROW_SE)
    elif angulo > 157.5 and angulo <= 202.5:
        display.show(Image.ARROW_S)
    elif angulo > 202.5 and angulo <= 247.5:
        display.show(Image.ARROW_SW)
```

```
elif angulo > 247.5 and angulo <= 292.5:  
    display.show(Image.ARROW_W)  
elif angulo > 292.5 and angulo <= 337.5:  
    display.show(Image.ARROW_NW)  
elif angulo > 337.5 and angulo <= 22.5:  
    display.show(Image.ARROW_N)
```

## Brújula indicando el norte

En este se intenta simplemente visualizar una aguja que apunte al norte

```
from microbit import *  
  
# Antes de nada calibrar  
compass.calibrate()  
  
# Mantener la aguja apuntando aproximadamente en la dirección correcta.  
while True:  
    sleep(100)  
    aguja = ((15 - compass.heading()) // 30) % 12  
    display.show(Image.ALL_CLOCKS[aguja])
```

Página extraída de Federico Coca [Guia de Trabajo de Microbit](#) CC-BY-SA

Revision #5

Created 27 September 2024 12:22:21 by Javier Quintana

Updated 1 October 2024 19:42:14 by Javier Quintana