Referenz Zumo32u4

LEDs (red, green, yellow)

```
ledRed(1); // LED einschalten
ledGreen(0); // LED ausschalten
ledYellow(0); // LED ausschalten
```

Taster

Objekt für den Taster erzeugen

```
Zumo32U4ButtonB buttonB;
```

Zustand des Tasters B abfragen

```
if (buttonB.isPressed()){
  …
}
```

Pausiert das Pogramm, bis Taster B gedrückt wurde

```
buttonB.waitForButton();
```

Batterie-Spannung

Gibt die aktuelle Batteriespannung zurück.

```
int spannung = readBatteryMillivolts();
```

Motoren

Objekt für die Motoren erzeugen

```
Zumo32U4Motors motors;
```

Geschwindigkeit einstellen (-400 … 400)

```
motors.setSpeeds(200, -200);
```

LCD-Display

Das LCD-Objekt erzeugen

```
Zumo32U4LCD lcd;
```

Löscht den Inhalt des Displays

```
lcd.clear();
```

Positioniert den Cursor (X: Zeichen, Y: Zeile)

```
lcd.gotoXY(0,1); // 1.Stelle, 2. Zeile
```

Text oder Daten auf dem Display anzeigen

```
lcd.print("Text"); // Text
lcd.print(wert);   // Daten
```

Zum Löschen von Buchstaben Leerzeichen nutzen

```
lcd.print("   "); // Leerzeichen
```

Liniensensoren initialisieren

Erzeugen des Objektes

```
Zumo32U4LineSensors lineSensors;
```

Initialisieren von 3 oder 5 Liniensensoren

```
lineSensors.initThreeSensors();
lineSensors.initFiveSensors();
```

Kalibrieren der Sensoren (messen und ggf. aktuell kleinsten oder größten Wert neu festlegen)

```
lineSensors.calibrate();
```

Liniensensoren auslesen

Zum Auslesen der Sensorwerte wird ein **Array** benötigt, in welchen die Werte übergeben werden

```
unsigned int senValues[5];
```

Die Sensorwerte können unkalibriert …

```
lineSensors.read(senValues);
```

… oder kalibriert ausgelesen werden

```
lineSensors.readCalibrated(senValues);
```

Die aktuelle Sensorwerte sind im Array abgelegt:

```
int leftSensor = senValues[0];
```

Position des Zumos über der Line ausgeben (die Sensorwerte im Array werden auch aktualisiert)

```
int pos = lineSensors.readLine(senValues);
```

Rad-Encoder (Odometry)

Das Objekt für die Encoder

```
Zumo32U4Encoders encoders;
```

Gibt die aufsummierten Encoder-Impulse zurück

```
int ticksLeft = encoders.getCountsLeft();
int ticksRight = encoders.getCountsRight();
```

Gibt ebenfalls die Anzahl der Impulse zurück, setzt den Wert anschließend aber auf null

```
int ticksLeft =
encoders.getCountsAndResetLeft();
int ticksRight =
encoders.getCountsEndResetRight();
```