Programmdokumentation

# Belegungen

## 

## Pins

|  |  |  |
| --- | --- | --- |
| **Pinname** | **Klasse** | **Funktion** |
| 3V3 | cNeigungssensor | Versorgung BNO055 |
| GND | cNeigungssensor | Ground |
| D2 | main | LED |
| D13 | CLenkersensor | Flanke A |
| D12 | CLenkersensor | Flanken B |
| D14 | CLenkersensor | Flanke M |
| D21 I2C-SDA | cNeigunssensor | Datenübertragung |
| D22 I2C-SCL | cNeigungssensor | Datenübertragung |
| D23 PWM | cGyroansteuerung | Motoransteuern\_Links |
| D19 PWM | cGyroansteuerung | Motoransteuern\_Rechts |
| D25 | cLenkersteuerung | PWML |
| D26 | cLenkersteuerung | PWMH |
| D32 | cLenkersteuerung | Phase |
| D33 | cLenkersteuerung | Current Sensor |
| D34 | cLenkersteuerung | Error 1 |
| D35 | cLenkersteuerung | Error 2 |

## 

## Speicher

|  |  |  |
| --- | --- | --- |
| **Speicherbereich** | **Klasse** | **Funktion** |
| Speicher: 0x0000 bis 0x00FF | cDatenlogger | Daten speichern |
| 0x00FF-0x0163 | cGyroansteuerung | Lookup |
| 0x??? - 0x??? | cNeigungssensor | Kalibrierungsdaten |

## Defines

|  |  |  |
| --- | --- | --- |
| **Variablenname** | **Klasse** | **Funktion** |
| DEFINE Gyro1PWMPin ?  DEFINE StandadGyroAnlauf 15  DEFINE maxLeistung 70  DEFINE verboteneZahl 6666 | cGyroansteuerung |  |
|  |  |  |
|  |  |  |

## Fehlercode (0 bis 255)

## 

|  |  |  |
| --- | --- | --- |
| **Fehlercode** | **Klasse** | **Beschreibung** |
| 0 |  | Alles okay |
| 1 |  |  |
| 2 |  |  |

## 

cNeigungssensor (TL & PK)

## Attribute:

## Operationen:

-Auslesen der Sensordaten (Gyroskop, Beschleunigungsmesser, Magnetometer) mit 100Hz Abtastfrequenz.

-Bereitstellung der Sensordaten.

-Speichern der Kalibrierung.

-Automatisches Laden der Kalibrierung

cLenkersensor (JS & TG)

cRegelung (wenn die Hardware steht)

cGyroansteuerung (AB + ML)

cLenkersteuerng (AB + ML)

cDrahtlos ( PK + TL)

cDatenlogger (JS + ML)

Alle getter werden gespeichert + Zykluszeit

main(TG + auf Rücksprache)

setup()

Alle nötigen Objekte erzeugen.

Anlaufprogramm: (Sensoren starten, Sensoren auslesen, Prüfung ist das Fahrrad in der richtigen Postion, wenn ja & Benutzerfrage dann Lenker kallibirieren, Benutzereingabe + alles klar, dann Gyros hochdrehen, warten auf Benutzereingabe, dann geht die Regelung los)

loop()

herunterfahren()