

# ADAPTIVES BREMSLICHT NACHRÜSTEN MIT ARDUINO

geschrieben von Peter S. | Dezember 27, 2019

## Zusammenfassung

In diesem Projekt wurde die Idee des adaptiven Bremslichts mit einem Arduino Nano (ATMEGA328P) Mikrocontroller neu interpretiert. Mit einem Transistor als Schalteinheit und einem Beschleunigungs- oder GPS-Sensor werden die Komponenten dem Bremssignal zwischengeschaltet – es muss nur ein zusätzliches Stromkabel gezogen werden. Jeder Fahrzeugtyp kann auf diese Weise einfach nachgerüstet werden, ohne dass eine komplizierte Verdrahtung oder eine Pedalkrafterfassung notwendig ist. Der Code wurde mit vorhandenen Arduino-Bibliotheken programmiert, die Gehäuse wurden in 3D gedruckt und die fertigen Module in der Praxis getestet. Der Gesamtpreis der benötigten Komponenten übersteigt nicht 15 \$.

**Achtung: Dieses Projekt ist für ein internationales Publikum geschrieben. Beachten Sie unbedingt die gesetzlichen Bestimmungen Ihres eigenen Landes, um festzustellen, ob eine solche Installation mit den geltenden Vorschriften vereinbar ist. Außerdem sind fortgeschrittene Kenntnisse auf dem Gebiet der Elektrotechnik erforderlich, um eine fachgerechte und sichere Installation zu gewährleisten. Im schlimmsten Fall kann das Bremslicht komplett ausfallen oder die Kabel können Feuer fangen, was in beiden Fällen zu immensen Schäden und Gefahren für Sie und andere führen kann. Die Autoren distanzieren sich hiermit vollständig von jeglicher Verantwortung! Handeln Sie auf eigene Verantwortung!**

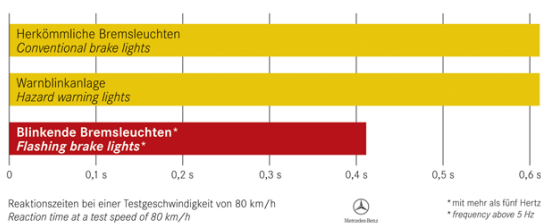
## Einleitung

Auffahrunfälle sind laut ADAC nach der Missachtung der Vorfahrt die zweithäufigste Unfallursache auf deutschen Straßen und gehen oft mit zu geringem Abstand zum Hintermann einher. Unachtsamkeit oder Fehleinschätzungen der Fahrer machen einen Unfall letztlich unausweichlich. Kollisionen führen oft zu schweren Verletzungen der Beteiligten im vorderen und hinteren Fahrzeug.

Wenn ein Fahrer im Notfall alles richtig macht und eine Vollbremsung durchführt, besteht trotzdem die große Gefahr, von einem nachfolgenden Fahrzeug in einen schweren Unfall verwickelt zu werden.

Mercedes-Benz (und mittlerweile auch einige andere Fahrzeughersteller wie BMW und Audi) begegneten diesem Problem mit einer intelligenten Lösung: Das adaptive Bremslicht. In dem Forschungsprojekt von Mercedes-Benz wurde die Anzahl der Auffahrunfälle mit 17% des gesamten Unfallgeschehens in Deutschland und 31% in den USA laut der National Highway Traffic Safety Administration (NHTSA) betitelt.

**Blinkende Bremsleuchten: Deutlicher Sicherheitsgewinn**  
*Flashing brake lights: a significant increase in safety*



Blinkende Bremsleuchten verkürzen die Reaktionszeit und sind eine wirksame Warnung vor Auffahrunfällen. Nachdem die Forscher verschiedene Setups getestet hatten (Herkömmliche Bremsleuchten vs. Warnblinkanlage vs. blinkendes Bremslicht), wurden die besten Ergebnisse hinsichtlich der Reaktion des hinteren Fahrers durch das Blinken des Bremslichts während einer Notbremsung erzielt. Der Bremsweg des Hintermannes konnte durch diese Methode bei einer Geschwindigkeit von 80 km/h um 4,4 m und bei 100 km/h um 5,5 m verkürzt werden, so ihre unveröffentlichte Studie.

Im folgenden Video wird die Technologie an einem Mercedes-Benz Modell demonstriert:



Grund genug, über eine Methode nachzudenken, wie dieses System auf einfache Art und Weise angepasst werden kann. Im Folgenden wurde versucht, mit möglichst einfachen Mitteln ein System zu bauen, um jedem Fahrer ein höheres Maß an Verkehrssicherheit zu bieten. Das hier beschriebene System kann leicht in jedem unspezifischen Fahrzeug nachgerüstet werden. Zunächst wurde eine Variante bestehend aus Mikrocontroller und Beschleunigungssensor entwickelt, die durch ein GPS-Modul ergänzt werden kann, wenn nicht nur die Beschleunigung, sondern auch die Geschwindigkeit in den Algorithmus einbezogen werden soll.

Ziel des Projektes war es auch, kosten-effizient zu arbeiten. Vorgestellt werden eine Variante, die nur auf einem Beschleunigungssensor basiert, mit insgesamt ca. 10 \$ Materialkosten, und eine Variante mit dem Zusatz eines GPS-Moduls mit ca. 15 \$ Materialkosten pro Stück.

## Materialien & Methoden

### Komponenten

- Microcontroller: ATMEGA328P (Arduino Nano Clone) (\$2.5)
- Acceleration-sensor: GY-521 MPU-6050 (\$1.5)
- GPS-sensor: GY-NEO6MV2 (\$6.15)
- Transistor: BD139 Genuine ON Semiconductor NPN 1.5 A / 80 V To-126 ZBDE W0HW (\$0.20) (Polarity here is C/E/B! Regular C/B/E polarity will of course work aswell)
- LEDs: T10 COB-LED (Socket depends on vehicle) (4x á \$0.97 = \$3.88)

### Software

- Autodesk Inventor Professional 2019 (Build: 265, Release: 2019.2)
- Fritzing beta (Ver. 0.9.4 )
- Arduino IDE 1.8.10 (Board: Arduino Nano, Processor: ATmega328P, Programmer: AVRISP mkII)
- Libraries for Arduino IDE:
  - without GPS: Wire.h
  - with GPS: Wire.h, Software.Serial.h, TinyGPS.h

### Methoden

Alle Gehäusekomponenten wurden mit PLA-Filament auf einem 3D-Drucker gedruckt. Die elektronischen Komponenten wurden mit 0,5 m<sup>2</sup> Litzenkabeln verlötet. Als Steckverbindungen wurden gecrimpte Dupont 2,64 mm Steckverbinder (auch bekannt als "Arduino-Kabel") verwendet.

Der Programmcode wurde über die USB-Schnittstelle in den Mikroprozessor geladen. Der in der Arduino-IDE implementierte serielle Plotter wurde zur Visualisierung und Kalibrierung der vom Beschleunigungssensor erfassten Signale verwendet.

### Quellcode

Das Funktionsprinzip des Quellcodes basiert auf einer Unterbrechung des elektrischen Signals zur Bremsleuchte mit Hilfe des Transistors. Das bedeutet, dass ohne Auslösen des Sketches (d. h. bei Bremsung bei langsamen Geschwindigkeiten oder nur geringen Beschleunigungsänderungen) der vom Bremspedal gesendete Strom durch den Mikrocontroller geleitet wird, um die Bremsleuchten zum Leuchten zu bringen. Erst sobald der Beschleunigungssensor eine Beschleunigung über einem definierten Schwellenwert und (falls verwendet) der GPS-Sensor eine Geschwindigkeit über einem definierten Schwellenwert erkennt, wird der Sketch ausgelöst und unterbricht das Dauerlichtsignal für eine definierte Zeit durch die Parameter BLINKON und BLINKOFF, wodurch ein Blinkeffekt entsteht. Ist der Sketch einmal aktiviert, hält das Blinken dann für eine definierte STOPPINGTIME an, unabhängig von den aktuell ausgelesenen Daten - so dass, wie im Beispielvideo, das Blinksignal bis zum Stillstand des Fahrzeugs oder für eine kurze Zeit während der starken Geschwindigkeitsreduzierung

anhält, um den gewünschten Warneffekt zu erzielen.

```
#include "Wire.h"
#include "SoftwareSerial.h"
#include "TinyGPS.h"

#define STOPPINGTIME 1500 // in ms
#define G_THRESHOLD 1.1 // activate above threshold
#define G 2.7e8 // = 1 g
#define BLINKON 75 // in ms
#define BLINKOFF 75 // in ms
#define INITFILT 400 // number of values averaged to get initial acceleration
#define FILT 140 // number of values averaged to get current acceleration

#define V_THRESHOLD 100 // in km/h
#define MAXFAILEDGPSBYTES 1000 // number of bytes received from gps to abort without valid
gps sequence
#define MAXTIMESINCELASTFIX 2000 // in ms

const int ledPin = 11;
const int ledPin2 = 13;
const int MPU_addr = 0x68; //I2C address
const int GPSrxPin = 3;
const int GPStxPin = 4;

bool blinkStatus, ledStatus, risingEdge;

unsigned long endTime, toggleTime;

double gforce, oldforce;

int16_t Xi, Yi, Zi;
int32_t Xc = 0, Yc = 0, Zc = 0;
double X, Y, Z, AccP;

int16_t Xbuf[FILT] = {0}, Ybuf[FILT] = {0}, Zbuf[FILT] = {0};
int bufpos = 0, i;
int32_t tmp;

SoftwareSerial gpsSerial(GPSrxPin, GPStxPin); //rx (D3),tx (D4) for GPS
TinyGPS gps; // create gps object
float speed = 0; //vehicle speed in km/h
long lat, lon;
unsigned long fix_age;
bool goodfix;

void setup() {
  //Connect to G-Sensor
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0); // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);

  Serial.begin(9600); // for USB monitor

  gpsSerial.begin(9600); // connect gps sensor
```

```

for (i = 0; i<INITFILT; i++) {
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,6,true); // request a total of 6 registers
  Xc+=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  Yc+=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  Zc+=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)

}

Xc = Xc/FILT;
Yc = Yc/FILT;
Zc = Zc/FILT;

pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, true);
pinMode(ledPin2, OUTPUT);
digitalWrite(ledPin2, true);
blinkStatus = false;
ledStatus = true;

}

void loop() {
  // GPS read
  i = 0;
  while(gpsSerial.available()) { //Data available?
    if(gps.encode(gpsSerial.read())) { //GPS telegram complete?
      speed = gps.f_speed_kmph();

      break; // end while loop
    }

    if (i++ > MAXFAILEDGPSBYTES) // cancel after MAXFAILEDGPSBYTES bytes without valid
    telegram
      break;
  }
  //check for valid signal
  gps.get_position(&lat, &lon, &fix_age);
  goodfix = ((fix_age == TinyGPS::GPS_INVALID_AGE) || (fix_age > MAXTIMESINCELASTFIX)) ? false
: true;

  //G-Sensor read
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,6,true); // request a total of 6 registers
  Xbuf[bufpos]=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  Ybuf[bufpos]=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  Zbuf[bufpos]=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)

  tmp = 0;
  for (i = 0; i<FILT; i++) {
    tmp += Xbuf[i];
  }
  Xi = tmp;
  tmp = 0;
  for (i = 0; i<FILT; i++) {
    tmp += Ybuf[i];
  }
}

```

```

Yi = tmp;
tmp = 0;
  for (i = 0; i<FILT; i++) {
    tmp += Zbuf[i];
  }
Zi = tmp;

X = Yc*Zi - Zc*Yi;
Y = Zc*Xi - Xc*Zi;
Z = Xc*Yi - Yc*Xi;

gforce = sqrt(X*X + Y*Y + Z*Z);

Serial.print(G_THRESHOLD * G);
Serial.print(',');
Serial.print(8e8);
Serial.print(',');
Serial.print(G);
Serial.print(',');
Serial.println(gforce);
delay(0);

// blinkstuff
risingEdge = false;

if (blinkStatus) {
  if (gforce > G_THRESHOLD * G) {
    endTime = millis() + STOPPINGTIME;
  }
} else {
  if ( (gforce > G_THRESHOLD * G) && ( !goodfix || (speed > V_THRESHOLD)) ) {
    risingEdge = true;
    blinkStatus = true;

    endTime = millis() + STOPPINGTIME;
  }
}

if (risingEdge) {
  toggleTime = millis() + BLINKOFF;
  digitalWrite(ledPin, false);
  ledStatus = false;
}

if (blinkStatus) {
  if (millis() > toggleTime) {
    if (ledStatus) {
      ledStatus = false;
      toggleTime = millis() + BLINKOFF;
      digitalWrite(ledPin, false);
      digitalWrite(ledPin2, false);
    } else {
      ledStatus = true;
      toggleTime = millis() + BLINKON;
      digitalWrite(ledPin, true);
      digitalWrite(ledPin2, true);
    }
  }
}

if (millis() > endTime) {
  blinkStatus = false;
}

```

```

    digitalWrite(ledPin, true);
    digitalWrite(ledPin2, true);
  }
}
}

```

## Ergebnisse

### Setup & Kalibrierung

Mikroprozessor und Beschleunigungsmodul wurden entsprechend dem Schaltplan verlötet (Abb. 1). Die Gehäuse wurden 3D-gedruckt, ein 3D-Rendering, wie die Komponenten darin eingebaut wurden, ist in Abb. 2 unten dargestellt.



**Abbildung 1:** Schaltplan der elektronischen Komponenten. Der Mikrocontroller fängt das Minus-Signal des Bremslichts ab und bewirkt hier eventuell eine Unterbrechung des Signals. Achtung! Der verwendete Transistor verwendet C/E/B-Polarität! Wird ein anderer Transistor verwendet, so ist dieser höchstwahrscheinlich Standard C/B/E-Polarität und muss entsprechend verdrahtet werden. Weitere Informationen entnehmen Sie bitte dem jeweiligen Datenblatt! (Fritzing)



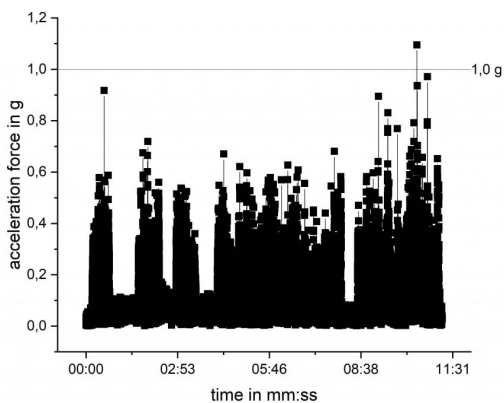
**Abbildung 2:** Elektronische Komponenten eingebettet in 3D-gedruckten Gehäusen (transparent). Vorne Arduino Nano mit Beschleunigungsmesser und Transistor, hinten der GPS-Sensor mit Antenne. (Autodesk Inventor Professional 2019)

Nach dem Hochladen des Programmcodes konnte der Beschleunigungssensor auch über den USB-Anschluss ausgelesen werden. Der serielle Plotter der Arduino-IDE machte es einfach, die Ausgabe in Echtzeit darzustellen. Für erste Tests war es ausreichend, das System nur über einen USB-Port zu betreiben. Da der USB-Port nur 5 V Versorgungsstrom liefert, wurde zu Testzwecken eine 5-V-LED in die Schaltung eingebunden, wie in Abb. 3 gezeigt.



**Abbildung 3:** Proof of Concept: Einfacher Testaufbau aus Mikrocontroller und Beschleunigungssensor mit USB-Stromversorgung und 5-V-LED. Die LED wird über den GPS 5 V Pin versorgt.

Es folgte eine Analyse der im Straßenverkehr auftretenden Beschleunigungskräfte und - um den Schwellenwert für eine Notbremsung zu ermitteln - wurden auch die (negativen) Beschleunigungen bei unterschiedlich starken Bremsvorgängen axial, sowie orthogonal zur Fahrtrichtung untersucht. Abb. 4 zeigt das Ergebnis einer Testfahrt in Form des Datenplots. Der Schwellenwert für eine Notbremsung wurde dabei auf 0,9 g festgelegt.



**Abbildung 4:** Beschleunigungsdiagramm der X/Y-Koordinaten während einer Testfahrt. Während normale Beschleunigungs- und Bremsvorgänge im Bereich um 0,5 g stattfinden, treten einzelne starke Bremsmanöver mit deutlich sichtbaren Spitzen um 1 g auf. (OriginLab Origin 2018)

### Test unter realen Bedingungen

Nach der Bestimmung des Schwellenwertes der Beschleunigungskraft einer Notbremsung mit 0,8 g wurde das System in einer 12-V-Umgebung an das Fahrzeugnetz angeschlossen und zwischen den Minuspol der Bremsleuchte geschaltet. Die Verdrahtung wurde wie in Abb. 1 gezeigt durchgeführt. Es werden zwei Stromkreise benötigt: Ein Dauerstrom, um den Arduino vom Einschalten der Zündung bis zum Ausschalten kontinuierlich mit Strom zu versorgen. Der zweite Stromkreis ist das Bremslicht selbst. Da hier nur dann Strom fließt, wenn auch die Bremse betätigt wird, ist dieser Stromkreis keine Dauerstromquelle und versorgt nur die Glühlampen mit Strom. Das Signal wird kontinuierlich durch den Mikrocontroller

geführt. Der Weg der Bremslichtschaltung durch den Mikrocontroller über den Eingang IN und den Ausgang Ground, der den Stromkreis schließt, führte ins Minus. Da es nicht möglich ist, die Verdrahtung im Fahrzeug zu zeigen, wurden die einzelnen Teile zu Demonstrationszwecken vorher so verdrahtet, wie sie später im Fahrzeug verdrahtet wurden. Der Aufbau ist in Abbildung 5 dargestellt. Wahlweise mit angeschlossenem GPS-Modul, oder nicht. Der Code ist zumindest so ausgelegt, dass er mit und ohne GPS-Modul funktioniert. Mit GPS-Modul ist eine Geschwindigkeitsschwelle aktiv, ohne Modul zählt nur der G-Wert. Es wurde eine Geschwindigkeitsschwelle von 80 km/h eingestellt. Die Tests waren erfolgreich, wie das Video (Abbildung 6) beweist.



**Abbildung 5:** Demonstration der Verkabelung für den Fahrzeugaufbau. Von links nach rechts: Der Mikrocontroller erhält 12 VDC aus dem Fahrzeugbordnetz. Der Minuspol des Bremslichts wird vom Mikrocontroller abgefangen, der auch eine Verbindung zur Masse herstellt. Alle Leuchten erhalten 12 VDC von aktivierten Bremslichtern. Der Anschluss des Bremslichtes wird nicht verändert, er liefert das "Signal" und führt Minus weg.

**Abbildung 6:** Video des Bremslichts bei verschiedenen Geschwindigkeiten (rechts). Auf der linken Seite ist ein Echtzeitdiagramm eingezeichnet, das die tatsächlichen g-Kräfte anzeigt. Die Schwelle ist auf 0,8 g (rote horizontale Linie) und 80 km/h eingestellt.

Während bei 40 km/h und Überschreiten des Schwellenwertes von 0,8 g keine Aktion ausgelöst wird und das Bremslicht normal leuchtet, wird oberhalb von 80 km/h die Skizze aktiviert und das Blinken eingeschaltet. Zur Kontrolle ist auch eine Bremsung oberhalb von 80 km/h, aber unterhalb des Schwellenwertes sichtbar. Wie erwartet löst das Programm hier nicht aus.

## Diskussion & Ausblick

In diesem Projekt wurde ein älteres Fahrzeug mit einem adaptiven Bremslicht nachgerüstet. Ab einem Schwellenwert der negativen Beschleunigung (g-Kraft beim Bremsen) bei Überschreiten einer definierten Geschwindigkeitsschwelle beginnt das Bremslicht bei Betätigung mit einer Frequenz unter 0,5 Hz zu blinken, um den nachfolgenden Verkehr auf gefährliches Bremsen hinzuweisen. Der Aufwand für Soft- und Hardware ist mit maximal 15 \$ überschaubar und stellt eine sinnvolle Investition für die Nachrüstung des eigenen Fahrzeugs dar - sofern die gesetzlichen Bestimmungen dies zulassen. Im Ernstfall können so schwere Wirbelsäulenverletzungen durch eine Kollision mit dem nachfolgenden Fahrzeug vermieden werden. Die Beschleunigungs- und Geschwindigkeitsdaten des Fahrzeugs können auch in anderen Projekten genutzt werden, z. B. zur Implementierung eines Diebstahlschutzes/GPS-Trackings.

## Referenzen

6211870 {6211870:UPS4496L},{6211870:9HGVHLTG} 1 apa 50 default 1437  
https://buildplate.de/wp-content/plugins/zotpress/  
%7B%22status%22%3A%22success%22%2C%22update%22%3Afalse%2C%22meta%22%3A%7B%22request\_last%22%3A0%2C%22request\_next%22%3A0%2C%22used\_cache%22%3Atrue%7D%2C%22data%22%3A%5B%7B%22key%22%3A%229HGVHLTG%22%2C%22library%22%3A%7B%22id%22%3A6211870%7D%2C%22meta%22%3A%7B%22creatorSummary%22%3A%22Daimler%20AG%22%2C%22parsedDate%22%3A%222019-12-28%22%2C%22numChildren%22%3A0%7D%2C%22bib%22%3A%22%26lt%3Bdiv%20class%3D%26quot%3Bcsl-bib-body%26quot%3B%20style%3D%26quot%3Bline-height%3A%202%3B%20padding-left%3A%201em%3B%20text-indent%3A-1em%3B%26quot%3B%26gt%3B%5Cn%20%20%26lt%3Bdiv%20class%3D%26quot%3Bcsl-entry%26quot%3B%26gt%3BDaimler%20AG.%20%282019%2C%20December%2028%29.%20%26lt%3Bi%26gt%3BAdaptive%20brake%20lights%3A%20Effective%20warning%20of%20rear-end%20collisions%26lt%3B%5C%2F%26gt%3B%20%5BService-Blog%5D.%20Daimler%20Media.%20%26lt%3Ba%20class%3D%26%23039%3Bzp-ItemURL%26%23039%3B%20href%3D%26%23039%3Bhttps%3A%5C%2F%5C%2Fmedia.daimler.com%5C%2FmarsMediaSite%5C%2Fen%5C%2Finstance%5C%2Fko%5C%2FAdaptive-brake-lights-Effective-warning-of-rear-end-collisions.xhtml%3Foid%3D9905287%26%23039%3B%26gt%3Bhttps%3A%5C%2F%5C%2Fmedia.daimler.com%5C%2FmarsMediaSite%5C%2Fen%5C%2Finstance%5C%2Fko%5C%2FAdaptive-brake-lights-Effective-warning-of-rear-end-collisions.xhtml%3Foid%3D9905287%26lt%3B%5C%2Fa%26gt%3B%26lt%3B%5C%2Fdiv%26gt%3B%5Cn%26lt%3B%5C%2Fdiv%26gt%3B%22%2C%22data%22%3A%7B%22itemType%22%3A%22webpage%22%2C%22title%22%3A%22Adaptive%20brake%20lights%3A%20Effective%20warning%20of%20rear-end%20collisions%22%2C%22creators%22%3A%5B%7B%22creatorType%22%3A%22author%22%2C%22firstName%22%3A%22%22%2C%22lastName%22%3A%22Daimler%20AG%22%7D%5D%2C%22abstractNote%22%3A%22%22%2C%22date%22%3A%2228.12.2019%22%2C%22url%22%3A%22https%3A%5C%2F%5C%2Fmedia.daimler.com%5C%2FmarsMediaSite%5C%2Fen%5C%2Finstance%5C%2Fko%5C%2FAdaptive-brake-lights-Effective-warning-of-rear-end-collisions.xhtml%3Foid%3D9905287%22%2C%22language%22%3A%22English%22%2C%22collections%22%3A%5B%22EF

