

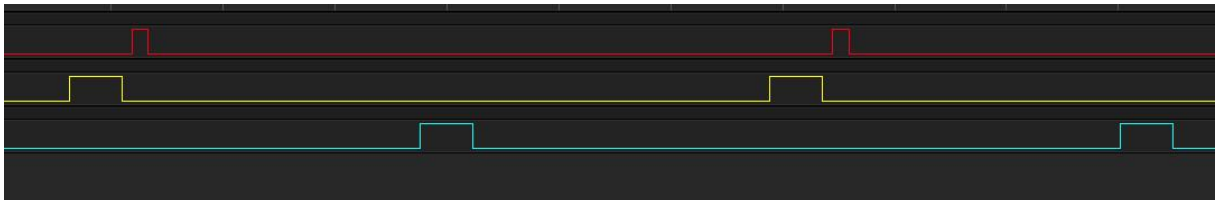
Elektronische Zündung mit ESP32

1. Firmware

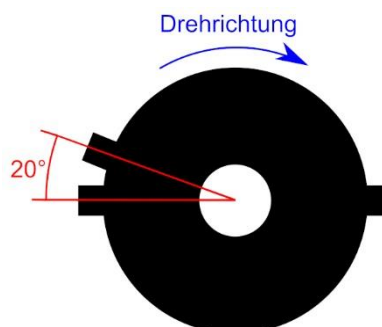
Die Steuerung erfolgt über einen Mikrocontroller ESP32. Drei der vorhandenen 8 Zählereinheiten werden genutzt. Alle drei Zähler erhalten ihren Takteingang auf Pin25. Der Takt wird ebenfalls im Microcontroller erzeugt und steht auf Pin26 zur Verfügung. Pin 25 und 26 müssen daher verbunden werden. Der verwendete Takt beträgt je nach Erfordernis 20 kHz oder 200 kHz. Bei höheren Drehzahlen wird der Takt auf 200 kHz umgeschaltet, um eine höhere Auflösung zu erhalten. Bei niedrigen Drehzahlen wird der Takt auf 20 kHz gesetzt, um ein Überlaufen der Zähler zu vermeiden. Über den Pin15 werden Sensorimpulse eingelesen. Die Sensorimpulse müssen so aufgearbeitet werden, dass der Logik High Pegel zwischen 2 V und 3.3 V liegt. Der Logik Low Pegel muss unter 1V liegen. Der Eingang reagiert auf die positive Flanke. Das Ausgangssignal zur Ansteuerung der Zündspulen liegt an den Pins 16 für Zündspule 1 und Pin 17 für Zündspule an. Der Logik High Pegel beträgt 3.3 V und ist mit maximal 10 mA belastbar. Welche Pegel für Zündspule „ein“ und „aus“ anliegen soll und auf welche Flanke der Sensor reagieren soll, kann im Sourcecode eingestellt werden.

```
//hardware dependent setup  
#define EDGE_RISING //active edge for the sensor input  
#define COIL_ON 1 //Ausgangssignal für die Zündspule  
#define COIL_OFF 0
```

Die Software kann einen oder drei Impulse pro Umdrehung verarbeiten. Bei einem Impuls wird die Zündspule 1 vor oder am Impuls ausgelöst. Die Zündspule 2 wird kurz vor oder genau nach einer halben Umdrehung ausgelöst. Rote Linie Sensor-Impulse, gelbe Linie Zündspule 1 und blaue Linie Zündspule 2.



Bei drei Impulsen ist einer für die Zündspule 1 und einer für die Zündspule 2. Der dritte Impuls ist ein Hilfsimpuls, um den Impuls für Zündspule 1 zu identifizieren. Zur Erzeugung der drei Impulse wird folgende Geberscheibe für einen induktiven Pickup Sensor verwendet.



Zuerst kommt der Hilfsimpuls dann der Impuls für Zündspule 1 und nach einer halben Umdrehung der Impuls für Zündspule 2.



Zur Konfiguration der Eckdaten und Steuerkurve können Profile erstellt werden, die im Flash-Filesystem des ESP32 abgespeichert werden. Ein Profil enthält die folgenden Parameter:

<i>name</i>	<i>Name des Profils, maximal 16 Zeichen</i>
<i>singlePulse</i>	<i>Ein Flag, wenn true Einzelimpulse wenn false Dreifachimpulse</i>
<i>maxRpm</i>	<i>Maximale Drehzahl</i>
<i>maxDwellTime</i>	<i>Maximale Einschaltdauer für die Zündspule in μs</i>
<i>minDischTime</i>	<i>Minimale Zeit zum Entladen der Zündspule in μs</i>
<i>correction1</i>	<i>Korrekturwert für Zündimpuls1 in Winkel $^{\circ}$</i>
<i>correction2</i>	<i>Korrekturwert für Zündimpuls2 in Winkel $^{\circ}$</i>
<i>curve</i>	<i>Tabelle mit zehn Wertepaaren bestehend aus Drehzahl in rpm und zugehörigem Zündwinkel in Winkel $^{\circ}$</i>

Der Arduino Sketch ist zur Übersichtlichkeit in vier Dateien aufgeteilt. In der Hauptdatei `esp32_zuendung_pcntz_bt_classic.ino` sind alle Definitionen und die Interrupt Verarbeitung für die Sensorimpulse. In der Datei `Bluetooth.ino` ist alles zur Kommunikation zwischen dem ESP32 und der Smartphone App. Die Datei `Pulsecount.ino` enthält die Initialisierung und die Interrupt Verarbeitung für die Zähler. Die Datei `profile.ino` enthält die Funktionen zum Laden, Auslesen und Speichern der Profile.

Der Zähler `STPCNT_UNIT` ermittelt im Einzelimpulsbetrieb die Drehzahl und den Basispunkt für den zweiten Zündimpuls nach einer halben Umdrehung. Im Dreifachimpulsbetrieb wird mit diesem Zähler ebenfalls die Drehzahl ermittelt. Zusätzlich sorgt er mit Hilfe des Zusatzimpulses für die Unterscheidung zwischen erstem und zweitem Zündimpuls.

Die beiden Zähler `ROTCNT_UNIT_1` und `ROTCNT_UNIT_2` sorgen für das exakte Ein- und Ausschalten der beiden Zündimpulse. Die dafür notwendigen Vergleichswerte werden in der Funktion `void setThreshold(pcnt_unit_t unit, int16_t ticks, uint16_t curRpm)` aus der aktuellen Drehzahl und Werten aus der Steuertabelle sowie den Korrekturwerten ermittelt.

Zum Kompilieren der Firmware sind neben dem ESP32 Package (V2.0.4 oder höher) die Bibliotheken `ArduinoJSON(V6.19.4 oder höher)` und `LittleFS_esp32(V1.0.6 oder höher)` erforderlich.

2. Android App

The screenshot shows the main interface of the 'Elektronische Zündung mit ESP32' app. At the top, there's a status bar with the time 9:51, data usage 0.9KB/s, and battery level 57%. Below the title bar, there are several icons for profile management: a wrench and screwdriver (load), a folder (save), a document with 'REN' (rename), a document (save), a document with a red X (delete), a document with a green arrow (import), and a document with a red arrow (export). A red dot indicates the 'aktuell geladenes Profil' (currently loaded profile). Below the icons, the current RPM is displayed as 'Drehzahl 960 RPM'. A blue line points to the 'aktuelle Drehzahl' (current RPM). Below the RPM display, there's a 'Profil' dropdown menu currently set to 'default'. To the right, there are radio buttons for '1 Impuls' (selected) and '3 Impulse'. Below these are fields for 'max. Drehzahl' (7500 rpm), 'max. Ladedauer' (4700 us), and 'min. Entladezeit' (2200 us). Further right, there are fields for 'Korr. 1' (0°) and 'Korr. 2' (0°). At the bottom, there's a table with 10 rows of data and a line graph showing the 'Kennlinie' (characteristic curve) of the ignition system.

Annotations:

- Profil umbenennen
- Profil unter neuem Namen speichern
- Profil exportieren
- Profil laden
- Profil speichern
- Profil löschen
- Profil importieren
- Zustand der Bluetooth Verbindung
- Konfigurations-Seite
- aktuell geladenes Profil
- Drehzahl 960 RPM
- aktuelle Drehzahl
- Profil default
- maximale Drehzahl
- 1 Impuls
- max. Drehzahl 7500 rpm
- maximale Einschaltzeit
- max. Ladedauer 4700 us
- min. Entladezeit 2200 us
- Umschaltung Impulszahl
- 3 Impulse
- Korr. 1 0°
- Korr. 2 0°
- Korrektur für Zündimpuls1
- Korrektur für Zündimpuls1
- minimale Entladezeit
- Kennlinie
- Tabelle mit zehn Einstellwerten
- grafische Darstellung der Tabellenwerte

Punkt	Drehzahl	Vorzündung
1	1400	5
2	1600	10
3	2000	17
4	2400	22
5	2900	25
6	3600	27
7	4300	28
8	6000	29
9	7400	29
10	8000	11

Mit dieser App, die derzeit nur für Android Smartphones funktioniert, kann in der elektronischen Zündung ein bestimmtes Profil geladen werden. Dieses Profil kann auch geändert und die Änderungen gespeichert werden. Die aktuelle Drehzahl wird angezeigt.

Die einzelnen Kommandos:

- Wechsel auf die Konfigurationsseite. Auf der Konfigurationsseite befindet sich ein Knopf, um eine Bluetooth Verbindung herzustellen und ein ausführliches Log. Mit dem Knopf fertig kehrt man zur Hauptseite zurück.
- Profil laden: Es erscheint eine Auswahl der verfügbaren Profile. Mit dem Knopf „Laden“ wird das ausgewählte Profil in der elektronischen Zündung geladen und in der App angezeigt.
- Profil umbenennen: Es kann ein neuer Name für das Profil gewählt werden. Das Profil „default“ kann nicht umbenannt werden.
- Profil speichern: Änderungen, die in der App gemacht wurden, werden an die elektronische Zündung übertragen und dort gespeichert.
- Profil unter neuem Namen speichern: Diese Funktion ermöglicht es neue Profile zu erstellen. Alle aktuellen Änderungen aus der App werden in der elektronischen Zündung unter dem neuen Namen gespeichert und dort auch als aktuelles Profil geladen. Existiert der neue Name bereits, so wird dieses Profil überschrieben.
- Profil löschen: Das aktuelle Profil wird in der elektronischen Zündung gelöscht und das Profil „default“ geladen. Das Profil „default“ kann nicht gelöscht werden.
- Profil exportieren: Das aktuell angezeigte Profil wird im Filesystem des Smartphones im JSON- Format gespeichert. (Noch nicht implementiert)
- Profil importieren: Ein Profil im JSON-Format kann aus einer Datei im Smartphone in die App importiert werden. Um das neue Profil zu speichern, muss das Profil als aktuelles Profil oder mit neuem Namen gespeichert werden. (Noch nicht implementiert)
- Anwendung verlassen: Das Anwendungsfenster wird geschlossen, die App aber nicht beendet.

Nach dem Start der App versucht diese eine Bluetooth-Verbindung zur elektronischen Zündung herzustellen. Ist dies nicht erfolgreich erscheint die Meldung „Zündung nicht gefunden“. In diesem Fall bleibt der Verbindungsstatus auf rot und die Funktionen sind gesperrt. Um einen neuen Verbindungsversuch zu starten, muss man auf die Konfigurationsseite wechseln und dort auf „Verbinden“ tippen. Konnte eine Verbindung hergestellt werden, wechselt der Verbindungsstatus auf grün.

Die Drehzahlanzeige wird alle 10 Sekunden aktualisiert.

Berechtigungen: Die App benötigt den Zugriff auf den Standort für Bluetooth erforderlich). Außerdem ist „Änderung der Bluetooth-Verbindung“ und „Dauerhafte Benachrichtigung“ erforderlich.