

# Übersicht

FreePZB ist eine Implementierung der Logik einer Punktförmigen Zugbeeinflussungsanlage basierend auf dem in der DB Richtlinie 483.0111 beschriebenen Verhalten. Die Implementierung ist in Lua geschrieben und macht selbst keine Annahmen über die Art der Simulation, in der sie eingesetzt wird. Sie ist ausgeführt als ein Datenobjekt, das über eine Funktion, die die aktuellen Daten erhält, aktualisiert wird. Damit lässt sie sich in beliebige Umgebungen einbinden.

Das komplette FreePZB befindet sich in der Datei freepzb.lua. Implementiert sind alle Zugarten, Geschwindigkeitskurven (als kontinuierliche Überwachung) und Überlagerungen.

Folgende Funktionen sind geplant, aber noch nicht implementiert:

- PZB Störbetrieb
- Überwachung der der Zugart entsprechenden Höchstgeschwindigkeit außerhalb von Beeinflussungen
- PZB Funktionsprüfung
- Einstellung der Höchstgeschwindigkeit mit Anpassen der 1000 Hz Kurve um einen (unrealistischen) Betrieb bei über 160 km/h zu ermöglichen

## Anwendung

Das Funktionsprinzip ist einfach, es wird über eine Initialisierungsfunktion ein Datenobjekt erzeugt, das mit einer Aktualisierungsfunktion regelmäßig mit aktuellen Daten aus der Simulation (Zeit, Geschwindigkeit, Beeinflussungen usw.) modifiziert wird. Aus dem Datenobjekt wird dann wiederum der aktuelle Zustand (Leuchtmelder, akustisches Signal, Zustand Bremsventil) ausgelesen und in die Simulation zurück geführt.

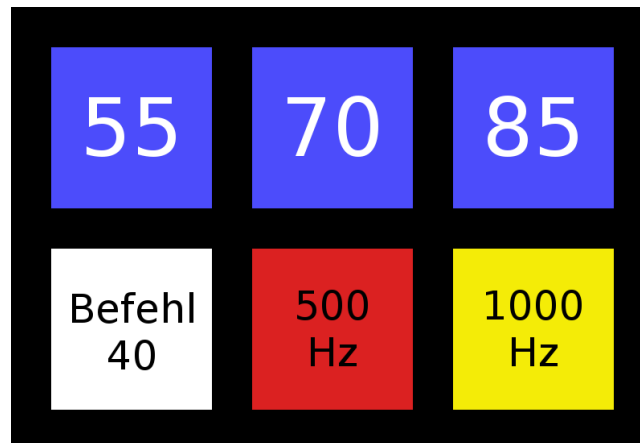
## Eingabedaten

Zugart	Die Zugart als String „O“, „M“ oder „U“. Nur bei der Initialisierung benötigt.
Zeit	Die aktuelle Simulationszeit in Sekunden. Dieser Wert muss positiv sein und darf ab der Initialisierung nicht rückwärts laufen. Bei der Initialisierung wird die aktuelle Zeit übergeben, so dass die Zeit des Simulators direkt verwendet werden kann, falls diese immer positiv ist.
Strecke	Die vom Triebfahrzeug zurückgelegte Strecke in Metern. Dieser Wert muss positiv sein und sollte nicht rückwärts laufen. Rollt das Triebfahrzeug rückwärts (gegenüber der mit dem Fahrtrichtungsschalter eingestellten Richtung) sollte dieser Wert entweder gleich bleiben oder entsprechend der Geschwindigkeit weiter steigen.
Geschwindigkeit	Die aktuelle Geschwindigkeit des Triebfahrzeugs in Kilometern pro Stunde. Dieser Wert muss positiv sein.
Magnet	Aktive Beeinflussung als Zahl entsprechend der Frequenz der Beeinflussung, 0 wenn keine Beeinflussung.
Tasten	Der Zustand der PZB Eingabetasten „Wachsam“, „Frei“, „Befehl“ und „Prüftaste“ als Lua Table mit den Feldern „wachsam“, „frei“, „befehl“ und „pruef“, jeweils mit einem Lua Boolean der angibt, ob die Taste betätigt ist (true) oder nicht (false).

Die durch aktive Magnete ausgelösten Aktionen passieren in dem Moment, in dem die Beeinflussung aufhört, so dass Dauerbeeinflussungen in dieser Implementation nicht passieren können. Es dürfen nur die aktiven Magnete übermittelt werden, die der mit dem Fahrtrichtungsschalter eingestellten Fahrtrichtung entsprechen, auch wenn das Triebfahrzeug rückwärts rollen sollte.

## Ausgabedaten

Die Ausgabe der Aktualisierungsfunktion besteht aus dem Zustand der Leuchtmelder und des akustischen Signalgebers sowie dem Zustand des PZB Bremsventils. Die 6 Leuchtmelder sind in Farbe, Beschriftung und Anordnung normalerweise in dieser Form anzutreffen:



FreePZB behandelt dabei die blauen Leuchtmelder als eine Gruppe, ebenso die 500/1000 Hz Beeinflussungs-Leuchtmelder, das heißt ein Wert der Ausgabe beschreibt jeweils den Zustand der ganzen Gruppe.

Das akustische Signal ist fahrzeugspezifisch, es kann ein Piepsen, Summen oder eine Sprachansage „Zugbeeinflussung“ sein.

Die PZB hat ein eigenes Bremsventil, mit dem sie eine Schnellbremsung auslösen kann. Dieses Ventil ist unabhängig von anderen Bremsventilen, das heißt die PZB kann selbstständig eine Schnellbremsung aufheben indem sie dieses Ventil schließt.

Die FreePZB Instanz ist eine Lua Table, die Ausgabe der Aktualisierungsfunktion befindet sich darin in den folgenden Feldern:

lampen_blaue	Zustand der blauen Leuchtmelder („55“, „70“, „85“), Werte siehe unten
lampen_magnet	Zustand der Beeinflussungs-Leuchtmelder („500 Hz“, „1000 Hz“), Werte siehe unten
lampe_befehl	Zustand der „Befehl 40“ Lampe als Boolean (true = an, false = aus)
akust_signal	Zustand des akustischen Signalgebers als Boolean (true = an, false = aus)
zwangsbremung	Zustand des Bremsventils als Boolean (true = offen, false = geschlossen)

Die möglichen Werte von lampen\_blaue sind Konstanten aus dem FreePZB Modul mit diesen Bedeutungen:

BLAU_AUS	Alle blauen Leuchtmelder sind dunkel
BLAU_55	„55“ an
BLAU_70	„70“ an

BLAU_85	„85“ an
BLAU_BLINK_55	„55“ blinkt
BLAU_BLINK_70	„70“ blinkt
BLAU_BLINK_85	„85“ blinkt
BLAU_WECHSEL	„70“ und „85“ blinken abwechselnd

Die möglichen Werte für `lampen_magnet` sind ebenfalls Konstanten mit diesen Bedeutungen:

MAGNET_AUS	Beide Leuchtmelder sind dunkel
MAGNET_1000	„1000 Hz“ an
MAGNET_500	„500 Hz“ an
MAGNET_ZWANG	„500 Hz“ und „1000 Hz“ blinken im Gleichtakt
MAGNET_STOER	„1000 Hz“ blinkt (tritt nur bei PZB Störbetrieb auf)

## **Initialisierung**

Mit der Funktion `create(zeit, strecke, zugart)` wird eine neue PZB Instanz erzeugt und zurück gegeben. Bei einem Fehler ( $\text{Zeit/Strecke} < 0$ , Zugart ungültig) wird `nil` zurück gegeben. FreePZB verwendet selbst keine globalen Variablen, alle Daten sind in der erzeugten Instanz enthalten.

Eine neu initialisierte Instanz wird beim ersten Einschalten der PZB und bei Fahrtrichtungswechsel verwendet. Das Startprogramm ist in dieser Instanz nicht aktiviert, so dass damit auch „fliegende Starts“ bei fahrendem Triebfahrzeug möglich sind ohne sofort zu einer Zwangsbremmung zu führen. Das ist etwa bei Einschalten während der Fahrt sinnvoll, oder etwa bei der Übergabe der Zugsicherung von einer LZB zur PZB.

Im Normalfall beim Einschalten der PZB im Stand muss das Startprogramm mit `activate_start(instanz, strecke)` aktiviert werden. Auch beim Umschalten des Fahrtrichtungsschalters aus der Mittelstellung auf Vorwärts sollte das Startprogramm (wieder) aktiviert werden.

## **Betrieb**

Im Betrieb wird die PZB Instanz regelmäßig mit aktuellen Daten aktualisiert mit der Funktion `update(instanz, zeit, strecke, geschwindigkeit, magnet, tasten)` und die oben beschriebenen Felder aus der Instanz gelesen, um Leuchtmelder etc. zu setzen.

Dieses Update ist am Besten mit jedem Simulationsschritt auszuführen, kann aber auch mit reduzierter Rate durchgeführt werden. Dann sollte das Update sofort bei Zustandsänderung der Magnete oder Tasten durchgeführt werden, ansonsten in regelmäßigen Abständen. Die Animation der Leuchtmelder ist im Allgemeinen dadurch nicht eingeschränkt, da die oben beschriebenen Zustände die nötigen Blinksequenzen abdecken und so unabhängig vom Update arbeiten.

Ausnahmen dabei sind die Bestätigung einer 1000 Hz Beeinflussung während der ersten 700 m einer vorherigen 1000 Hz Beeinflussung wobei der „1000 Hz“ Leuchtmelder für eine halbe Sekunde dunkel geschaltet wird, der Funktionstest (noch nicht implementiert) bei dem einzeln durch die Leuchtmelder geschaltet wird sowie das Ende einer restriktiven Überwachung wobei ein Signalton für eine halbe Sekunde gegeben wird. Am Besten sollte das Update daher mindestens 5 mal pro (Simulations-)Sekunde ausgeführt werden.