

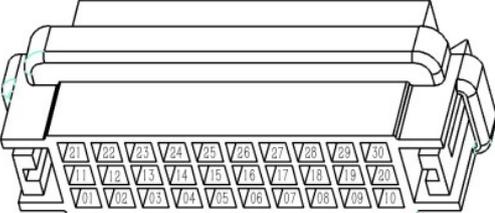
Geschwindigkeits-Begrenzer für den Lingbo LMCB HK2AE "Holländer-Version" : Details

(jolep, 20.03.2025)

Das Dokument "Doku Speed-Limit Inoa Sli5.pdf" enthält die Übersicht für den Aufbau und muss zum Verständnis hier ebenfalls mit beachtet werden.

Die Schaltung im Lingbo LMBC HK2AE Controller

Der 30 polige Stecker hat die folgenden relevanten Pins, hier rot markiert



Pin	Function	Voltage range	Pin	Function	Voltage range
1	HALL Sensor A	0~3.3V	16	GND	0V
2	Low Gear Input	0~5V	17	Reserve Input	0~5V
3	High Gear Input	0~5V	18	Cruise Input	0~5V
4	HALL Sensor 5V	5V	19	Wheel Sensor Alarm	0~B+
5	Single-Stand 5V or CANL ¹	0~5V	20	Anti theft Device ACC	B+
6	HALL Sensor GND	0V	21	Brake(Low Active)	0~5V
7	Anti theft Device GND	0V	22	HALL Sensor C	0~3.3V
8	Reverse Input	0~5V	23	RX/485B	0~5V
9	Wheel Sensor Output	0~B+			
	or Motor Temperature ²	0~3.3V	24	Reserve Input	0~5V
10	ACC	B+	25	Reserve Input	0~5V
11	Brake(High Active)	0~12V	26	Throttle GND	0V
12	HALL Sensor B	0~3.3V	27	Throttle Signal	0~5V
13	TX/485A	0~5V	28	Throttle 5V	5V
14	Anti theft Signal	0~5V	29	GUI 3.3V	0~3.3V
15	Reserve Input or CANH ¹	0~5V	30	Anti theft Device B+	B+

B+ is the battery voltage; Note 1: limit for HK3AC and HJ3AC; Note2: limit for HK3AC and HJ3AC.

Bild 1: Auszug aus dem Datenblatt des Lingbo Controllers.

"Hall-Sensor A" kommt aus dem Motor und zeigt (zusammen mit Hall B und C) dem Controller die Phasenlage der Spulen und Magnete.

"Throttle Signal" ist der Eingang vom "Gasgriff".

Verfolgt man diese Signale auf der Controller-Platine, bekommt man folgendes raus

- Hall A geht über ein paar Bauteile an den Hauptcontroller (STM32) Pin 38
- Throttle Signal geht über ein paar Bauteile auch an den STM32 an Pin 13

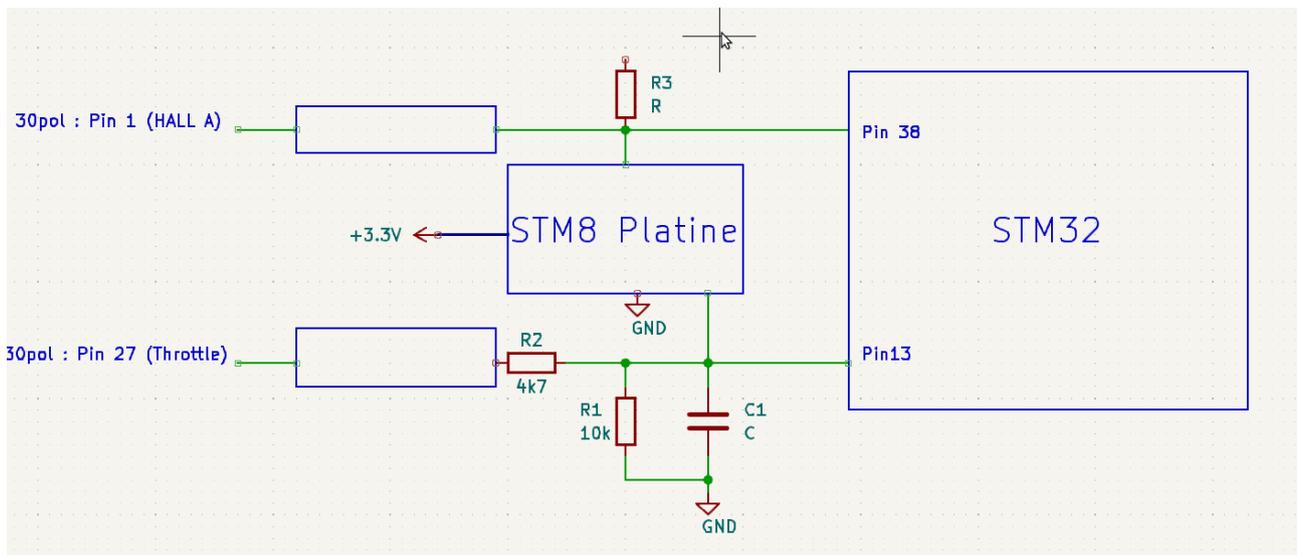


Bild 2: der relevante Teil des Controllers. Der STM32 ist der zentrale Microcontroller des Lingbo LMBC HK2AE. Die blauen Boxen links sind nicht komplett bekannte Koppel- und Schutznetzwerke, wichtig ist aber nur, was am STM32 ankommt.

Messungen am Controller

Ein paar Messungen zeigen :

- Hall A ist 0 bis 3,3V, wird zu ca. 0,8 bis 3,3 V am STM32. Dieses Signal zeigt 30 steigende Flanken bei einer Umdrehung des Rades
- Throttle Signal ist 0,9V ("Leerlauf") bis 4,2V ("Vollgas"), also nach dem Spannungsteiler ca. 0,6 bis 2,8V

Damit ist klar : misst man die Frequenz oder Periodendauer an Hall A, dann weiß man die Geschwindigkeit. Den Radumfang habe ich zu 1,41 m gemessen. Also bekommt man alle 4,7 cm eine steigende Flanke von Hall A. Die Frequenz ist also

$$f = v(\text{km/h}) / (3,6 * 0,047 \text{ m})$$

Eine Geschwindigkeit von 45 km/h führt zu 266 Hz an Hall A das ergibt eine Periodendauer p von 3,76 ms.

oder umgekehrt

$$v (\text{km/h}) = 3,6 * 0,047 / p$$

Die Idee

Steigt diese Frequenz (oder fällt die Periodendauer) in den Bereich der gewünschten Abregelung, dann reduziert man die Spannung am Throttle Signal und nimmt dadurch quasi "Gas" weg.

Das kann ein kleiner Microcontroller mit dem folgenden Aufbau problemlos übernehmen:

- ein STM8 misst die an PB4 anliegende Frequenz bzw. Periodendauer und
- gibt abhängig von diesem Wert an PC5 ein pulsweitenmoduliertes Signal (PWM) aus. R1 / C1 glätten dieses, und D3 führt es zum Throttle-Eingang des Hauptcontrollers. Dieses Signal kann, wegen dem 4,7 kOhm Widerstand auf dem Controller, das vom "Gasgriff" kommende Signal also übersteuern, aber nur in Richtung kleinerer Werte, damit also die Geschwindigkeit reduzieren

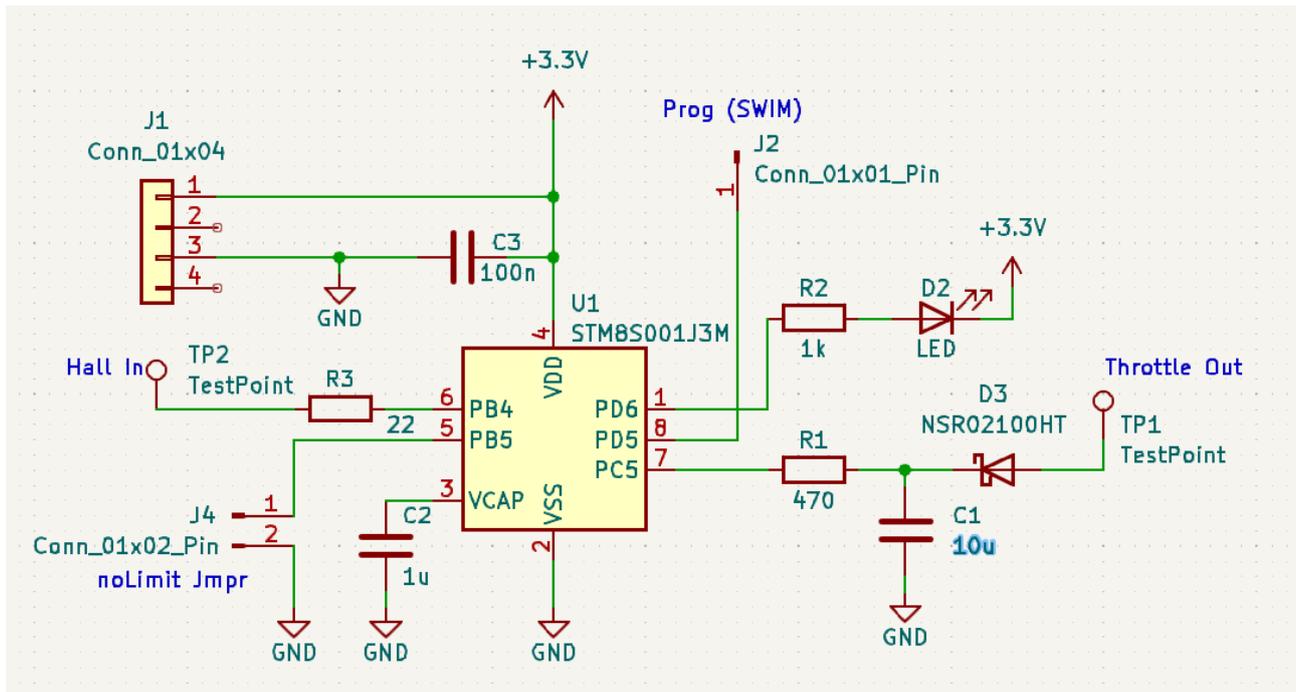


Bild 3 : Stromlaufplan

Ein paar Hinweise zum Stromlauf:

- die LED zeigt an, wenn die Regelung aktiv ist und nicht am oberen oder unteren Anschlag läuft. Sie ändert dann alle 50 ms ihren Zustand
- J1 wird zur Montage verwendet und speist 3,3V und GND zur Platine.

Algorithmus

- jede steigende Flanke an PB4 erzeugt einen Interrupt. Es wird Zähler 1 hochgezählt und ein interner, freilaufender Zähler (Timer 4, Auflösung 32 µs) gespeichert und dessen Wert von der vorhergehenden Flanke subtrahiert. Damit erhält man die Periodendauer.
- Alle 50 ms wird nun ausgewertet und eingestellt:
 - Ist eine Steckbrücke (Jumper) zwischen dem gelben und schwarzen Pin gesteckt (siehe folgendes Bild der Platine), so ist die Regelung außer Betrieb und der Roller maximal schnell.
 - Ist Zähler 1 kleiner 11, ist der Roller langsamer als 33 km/h, der PWM wird auf maximale Spannung gestellt. Die Diode ist dann in Sperrrichtung, der "Gasgriff" ist unbeeinflusst von der Schaltung.
 - Ist Zähler 1 größer, dann wird die Summer der 4 letzten Periodendauern ausgewertet (damit bekommt man eine höhere Auflösung und ruckelfreies fahren)
 - Ist die durchschnittliche Periodendauer größer als 3,9 ms (v langsamer als 43,3 km/h), wird der PWM immer noch auf maximaler Spannung gehalten.
 - Darunter beginnt die Abregelung in 85 Stufen. Die ersten ca. 20 Stufen sind ohne Auswirkung, weil die Spannung am Knoten R1/C1/D3 und der Abfall über die Diode noch zu hoch bleibt und daher kein Strom fließt.
 - Ist die Periodendauer kürzer als 3,2 ms (v größer als ca. 52,3 km/h), erreicht der PWM sein Minimum (15%), die Spannung am Knoten R1/C1/D3 erreicht 495 mV, die Diode zieht das "Gas"-Signal maximal nach unten.
 - Es sind also ca. 65 Stufen am "Gas"-Signal messbar, je nach Periodendauer.
 - Zusammengefasst : zwischen 45,5 km/h und 52,8 km/h erfolgt eine lineare Reduktion der Spannung am Knoten R1/C1/D3 und damit eine lineare "Gas"-Reduktion.
 - Dabei stellt sich eine Maximalgeschwindigkeit von (nach Tacho) 52-53 km/h ein, was in Realität ca. 47-48 km/h entspricht.

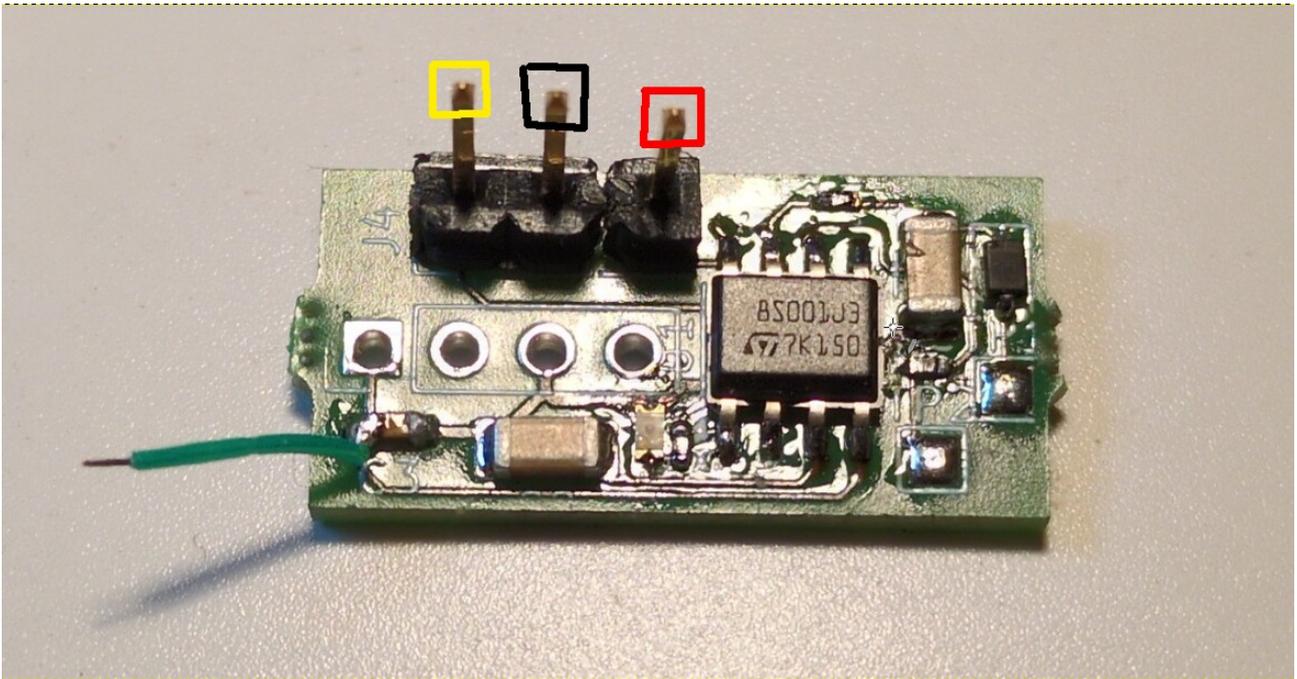


Bild 4 : Platine

Die 3 Pins haben folgende Funktion:

- schwarz : GND.
- gelb : Brücke nach GND schaltet die Regelung komplett ab
- rot : Programmierung über z.B. ST-Link V2 (Signal SWIM)

Hinweise

- aufgebockt bekommt man ein Schwingen der angezeigten Geschwindigkeit zwischen ca. 50 und ca. 54 km/h. Dieses war bei den ersten Versuchen beim Fahren ebenfalls zu spüren. In der aktuellen Version ist es unter Last quasi nicht mehr spürbar.
- Ein verbesserter Algorithmus könnte auch dieses Schwingen noch entfernen, aber das übersteigt meine Programmierfähigkeiten.
- Die Software ist ein C-Programm. Als Compiler verwende ich "sdcc". Download läuft z.B. über einen ST-Link V2 und den roten Pin im Bild oben.