

Diplomarbeit

Höhere Technische Bundeslehr- und Versuchsanstalt Salzburg
Abteilung für Elektrotechnik

Entwicklung eines emissionsfreien Sportmotorrades

Entwicklung der Zentralsteuerung / Projektleitung

Martin Kronberger 5AHET Betreuer: Dipl.-Ing. (FH) Johannes Ferner

Entwicklung des Antriebssystems

Jakob Lackner 5AHET Betreuer: Prof. Dipl.-Ing. MBA Adolf Reinhart

Entwicklung des Akkusystems

Simon Kern 5AHET Betreuer: Prof. Dipl.-Ing. Reinhold Benedikter

Entwicklung der mechanischen Komponenten

Tobias Schmeisser 5AHET Betreuer: Prof. Dipl.-Ing. Peter Lindmoser

Eidesstaatliche Erklärung

Wir erklären an Eides statt, dass wir die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht haben. Wir versichern, dass wir dieses Diplomarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin oder einem Beurteiler) in irgendeiner Form als Prüfungsarbeit vorgelegt haben.

Gendererklärung

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums angewendet. Es wird an dieser Stelle darauf hingewiesen, dass die ausschließliche Verwendung der männlichen Form geschlechtsunabhängig verstanden werden soll.

Martin Kronberger

Ort, Datum

Jakob Lackner

Ort, Datum

Simon Kern

Ort, Datum

Tobias Schmeisser

Ort, Datum

Vorwort

In immer mehr Großstädten werden Fahrzeuge mit Verbrennungsmotoren verboten. Viele Motorräder und Autos können nicht mehr produziert werden, da sie die immer strenger werdenden Abgasnormen nicht mehr einhalten können und das Thema der Klimaerwärmung wird immer präsenter und immer mehr Menschen versuchen ihren „carbon footprint“ zu verkleinern.

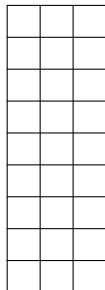
Doch leider gibt es für Motorradfahrer zumeist keine wirklichen Möglichkeiten, um für ihr Hobby auf eine emissionsfreie Alternativen umzusteigen. Denn zumeist ist das Preis-Leistungsverhältnis, oder auch das Produkt selbst, nicht sehr einladend. Daher ist unser Ziel die Entwicklung in diesem Bereich voranzutreiben und dadurch den Markt zu vergrößern, wodurch immer mehr und bessere Produkte angeboten werden können.

Danksagung

TEXT DANKSAGUNG

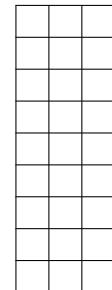
DIPLOMARBEIT

DOKUMENTATION



DIPLOMA THESIS

DOCUMENTATION



Erklärung

Die unterfertigten Kandidaten haben gemäß §34 (3) SchUG im Verbindung mit §22 (1) Zi. 3 lit. b der Verordnung über die abschließenden Prüfungen in den berufsbildenden mittleren und höheren Schulen, BGBl. II Nr. 70 vom 24.02.2000 (Prüfungsordnung BMHS), die Ausarbeitung einer Diplomarbeit mit der umseitig angeführten Aufgabenstellung gewählt. Die Kandidaten nehmen zur Kenntnis, dass die Diplomarbeit in eigenständiger Weise und außerhalb des Unterrichtes zu bearbeiten und anzufertigen ist, wobei Ergebnisse des Unterrichtes mit einbezogen werden können. Die Abgabe der vollständigen Diplomarbeit hat bis spätestens

03.04.2020

beim zuständigen Betreuer zu erfolgen. Die Kandidaten nehmen weiters zur Kenntnis, dass gemäß §9 (6) der Prüfungsordnung BMHS nur der Schulleiter bis spätestens Ende des vorletzten Semesters den Abbruch einer Diplomarbeit anordnen kann, wenn diese aus nicht beim Prüfungskandidaten / bei den Prüfungskandidaten gelegenen Gründen nicht fertiggestellt werden kann.

Kandidaten / Kandidatinnen	Unterschrift
Martin Kronberger	
Jakob Lackner	
Simon Kern	
Tobias Schmeisser	

Prof. Dipl.-Ing. Reinhold Benedikter
Prüfer

Prof. Dipl.-Ing. (FH) Johannes Ferner
Prüfer

Prof. Dipl.-Ing. Adolf Reinhart, MBA
Prüfer

Prof. Dipl.-Ing. Peter Lindmoser
Prüfer

Prof. Dipl.-Ing. (FH) Roland Holzer
Abteilungsvorstand

Dipl.-Ing. Dr.tech. Franz Landertshamer
Direktor

Inhaltsverzeichnis

I Einführung	2
1 Projektteam	2
2 Projektbetreuer	3
3 Aufgabeneinteilung	3
II Einleitung	5
1 Motivation	5
2 Zielsetzung	5
3 Topologie des Gesamtsystems	5
4 Leitfaden	5
III Stand der Technik	6
1 Steuereinheiten	6
1.1 Raspberry PI	6
2 Bussysteme	6
2.1 SPI Bus	6
2.2 CAN Bus	6
IV Mechanische Umsetzung	7
V Human-Computer Interaction System	8
1 Übersicht	8
1.1 Grundfunktionen des Systems	8
1.2 Steuereinheit	9
1.3 Grundaufbau des Systems	9
2 Spannungsversorgung	10
2.1 Aufbau des Versorgungssystems	10
2.1.1 12V Versorgungssystem	10
2.1.2 5V Versorgungssystem	10
2.1.3 Abschalten der Spannungswandler	10
3 Steuerung der Peripherie	11
3.1 Hardware	11
3.1.1 Input	11
3.1.2 Output	12
3.2 Software	12
3.2.1 gpiozero	12
3.2.2 threading	12
4 Benutzeroberfläche	13
4.1 Hardware	13
4.1.1 Befestigung	14
4.2 Software	14
4.2.1 Aufbau	14
4.2.2 Nutzer / Berechtigungen	15
4.3 Komponenten	15
4.3.1 Navigationsmenü	15
4.3.2 Balken Anzeige	16

4.3.3	Modus Anzeige	16
4.3.4	Graph	17
4.3.5	Weitere Komponenten	17
4.4	Programm Fenster	18
4.4.1	Login	18
4.4.2	Fahrdaten	18
4.4.3	Akku- und Ladedaten	19
4.4.4	Fahrdaten Diagnose	19
4.4.5	Fehler	20
4.4.6	Nutzer und Berechtigungen	20
4.5	Realisierung der Benutzeroberfächer	21
4.5.1	QML	21
4.5.2	Qt-Quick	21
4.5.3	Slots und Signals	21
4.5.4	Bridge	22
5	Kommunikation	23
5.1	Hardware	23
5.1.1	CAN-Modul	23
5.1.2	Netzwerkstruktur	23
5.2	Listener	24
5.2.1	Konfigurieren der Schnittstelle	24
5.2.2	Empfangen der Daten	24
6	Fahrdatenspeicher	25
6.1	Datenbankstruktur	25
6.1.1	Benutzer System	25
6.1.2	Motor Daten	25
6.1.3	Fehler Tabelle	27
6.1.4	Akku Daten	27
6.2	Handler	27
6.2.1	Konfigurieren der Schnittstelle	27
6.2.2	SELECT Befehl	28
6.2.3	INSERT Befehl	28
	VI Antriebsstrang	29
	VII Akku und Ladekonzept	30
	VIII Hintergebnis	31
A Arbeitsnachweis		32
1	Zeitplan	32
2	Kosten	32
B Programmcode		33
C CAD-Zeichnungen		34
D Schaltpläne		35
E Datenblätter		36
0.1	Mean Well RSD-30H-5	37
0.2	Mean Well SD-350C-12	47
0.3	Raspberry Pi 4 Moddel B	50
Literaturverzeichnis		63
Abbildungsverzeichnis		63
Tabellenverzeichnis		63

Codeverzeichnis**64**

Kapitel I

Einführung

1 Projektteam



Martin Kronberger



Jakob Lackner



Simon Kern



Schmeisser Tobias

2 Projektbetreuer

Prof. Dipl.-Ing. Reinhold Benedikter

unterstützte Jakob Lackner bei der Entwicklung des Akku- und Ladesystems

Prof. Dipl.-Ing. (FH) Johannes Ferner

unterstützte Martin Kronberger bei der Entwicklung des Human-Computer Interaction Systems

Prof. Dipl.-Ing. Adolf Reinhart, MBA

unterstützte Jakob Lackner bei der Entwicklung des Antriebssystems

Prof. Dipl.-Ing. Peter Lindmoser

unterstützte Tobias Schmeisser bei der Entwicklung der mechanischen Komponenten

3 Aufgabeneinteilung

Martin Kronberger

- Projektleitung
- Projektfindung und Projektplanung
- Projektaufteilung
- Erstellen der Einreichdokumente
- Entwickeln der Hardware des Human-Computer Interaction Systems
- Entwickeln der Software des Human-Computer Interaction Systems
- Planung und Umsetzung der elektrischen Installation
- Verfassen der Dokumentation

Jakob Lackner

- Projektfindung und Projektplanung
- Projektaufteilung
- Entwicklung des Antriebssystemes
- Entwicklung der Software des Motorsteuergerätes
- Erstellen der Einreichdokumente
- Verfassen der Dokumentation

Simon Kern

- Projektfindung und Projektplanung
- Projektaufteilung
- Entwicklung des Akkusystems
- Erstellen der Einreichdokumente
- Verfassen der Dokumentation

Tobias Schmeisser

- Projektfindung und Projektplanung
- Projektaufteilung
- Entwicklung der mechanischen Komponenten
- Entwicklung der Getriebemechanik
- Erstellen der Einreichdokumente
- Verfassen der Dokumentation

Kapitel II

Einleitung

- 1 Motivation**
- 2 Zielsetzung**
- 3 Topologie des Gesamtsystems**
- 4 Leitfaden**

Kapitel III

Stand der Technik

1 Steuereinheiten

1.1 Raspberry PI

2 Bussysteme

2.1 SPI Bus

2.2 CAN Bus

Kapitel IV

Mechanische Umsetzung

Kapitel V

Human-Computer Interaction System

1 Übersicht

Das Human-Computer Interaction System ist, wie der Name schon sagt, die Komponente, welche als Schnittstelle zwischen dem Nutzer und dem gesamten elektrischen System dient. Durch es sollte die fehlerfreie Nutzung der Funktionen des Motorrades gewährleistet sein. Ebenso sollte es wichtige Fahrdaten und andere Informationen speichern und dem User anzeigen können. Wichtig ist das System, trotz der großen Komplexität, so intuitiv und nutzerfreundlich wie möglich zu gestalten.

1.1 Grundfunktionen des Systems

Die geplanten Funktionen des HCIS¹ lassen sich grob in vier Grundfunktionen einteilen.

- **Steuerung der Peripherie**

Die Schalter und Buttons am Lenker, welche zuvor über den Kabelbaum die Leuchten, Blinker und die Hupe gesteuert haben, werden nun über die General-purpose input/output (GPIO) des Raspberry Pi Mikrocomputers gesteuert.

- **Graphische Benutzeroberfläche**

Dient der Anzeige wichtiger Fahr- und Ladedaten, welche entweder in Echtzeit oder über die Datenbankschnittstelle abgerufen und graphisch angezeigt werden können.

- **Kommunikation mit den Steuereinheiten des Motorrades**

Über CAN-Bus werden Daten von dem Batterie Management Systems (BMS) und der Curtis Motorsteuerung empfangen und an die Benutzeroberfläche zur Anzeige und an die Datenbankschnittstelle zur Langzeitsicherung der Fahrdaten weiter gegeben.

- **Speichern der relevanten Fahrdaten über die Datenbankschnittstelle**

Die über den CAN-Bus empfangenen Daten werden sofort an die Datenbankschnittstelle (Handler) weitergegeben um die Daten für Datenauswertung und Testberichte zu speichern. Ebenso bezieht das Diagnosesystem der Benutzeroberfläche die Daten über diese Schnittstelle.

¹Abkürzung: Human-Computer Interaction System

1.2 Steuereinheit

Als Basis zur Auswahl der Steuereinheit wurden die zuvor erläuterten Grundfunktionen herangezogen genommen. Die Ausgewählte Steuereinheit sollte diese erfüllen können und ebenso Potential zur Erweiterung der Funktionen bieten. Genauso wichtig war das eine große Flexibilität und Individualität erreicht werden kann, um nicht in der Umsetzung unserer Ideen eingeschränkt zu sein. Zur Auswahl standen verschiedene Speicherprogrammierbare Steuerungen und Mikrocomputer, doch letzten Endes überzeugte der Mikrocomputer Raspberry Pi.

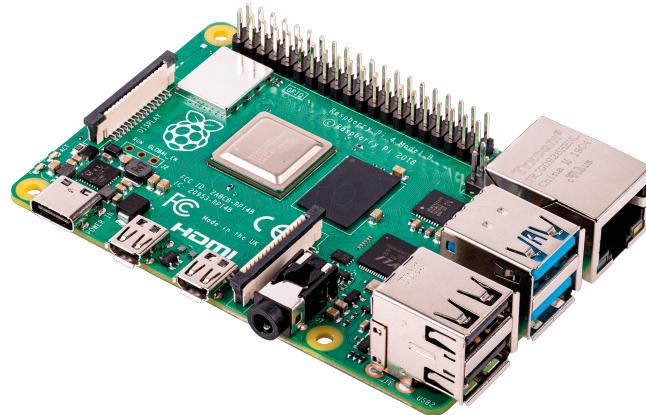


Abbildung V.1: Raspberry Pi - Steuereinheit des HCIS

1.3 Grundaufbau des Systems

In der Abbildung V.2 wird der Grundaufbau des Systems und die Datenverbindungen der folgenden Komponenten veranschaulicht.

- Raspberry Pi - Die Steuereinheit des Systems.
Kommuniziert über CAN-Bus mit den anderen Steuerkomponenten des Motorrades.
- User Input - Die vorhandenen Schalter am Lenker des Motorrads werden direkt mit den Eingängen des Raspberry Pis verbunden.
- Peripherie - Die Grundkomponenten des Motorrades wie Scheinwerfer oder Hupe. Diese werden über Relais, welche an die Ausgänge des Raspberry Pis angeschlossen sind, gesteuert.
- Dashboard - Der Bildschirm zur Anzeige der verarbeiteten Informationen. Dieser wird über HDMI und USB mit dem Raspberry Pi verbunden.

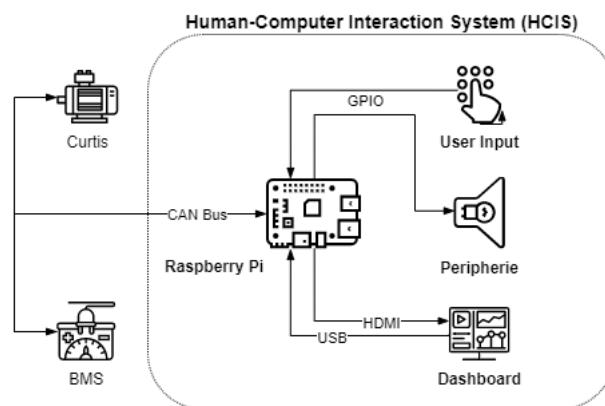


Abbildung V.2: Grundaufbau des Human-Computer Interaction Systems

Nicht in der Abbildung dargestellt ist die Versorgung der einzelnen Komponenten, welche in dem folgenden Abschnitt noch genauer erläutert wird.

2 Spannungsversorgung

2.1 Aufbau des Versorgungssystems

Das Versorgungssystem des Motorrades besteht aus zwei Spannungsebenen: Einer 12V Ebene zur Versorgung der Peripherie des Motorrades und einer 5V Ebene, welche nur den Raspberry Pi und seine Komponenten beinhaltet. Diese Ebenen werden durch DC-DC Wandler erzeugt, welche direkt an den Akku des Motorrades angeschlossen werden.

Wichtig hierbei ist, dass alle ausgewählten Spannungswandler über einen Kurzschluss- und Überstrom-Schutz verfügen. Dies macht es uns möglich diese Versorgungssysteme, solange die Drähte auch nach dem maximalen Strom der Spannungswandler dimensioniert wurden, ohne jegliche Leistungs- und Überstrom-Schutzorgane aufzubauen. Die Spannungswandler schalten bei jeglichen Fehlern ab und verbrennen die überschüssige Leistung über einen eingebauten Widerstand. Sobald der Fehler behoben wurde, schalten sich die Spannungswandler automatisch wieder ein.

2.1.1 12V Versorgungssystem

Um den Spannungswandler dimensionieren zu können mussten vorher alle Bauteile, welche über die 12V versorgt werden sollten, zusammengefasst werden, um die mindestens benötigte Leistung des Spannungswandlers zu errechnen.

Bauteilbezeichnung	Spannung	Leistung
Tagfahrlicht	12V	10W
Abblendlicht	12V	10W
Aufblendlicht	12V	20W
Hupe	12V	10W
Rücklicht	12V	21W
Kennzeichenbeleuchtung	12V	5W
Blinker links	12V	2 x 10W
Blinker rechts	12V	2 x 10W
Bildschirm	12V	12W
Gesamt	12V	128W

Tabelle V.1: Berechnung der Leistung des 12V-Systems

Der Spannungswandler wurde nun nach der größt möglichen Leistung, welche auftritt wenn alle Bauteile gleichzeitig auf Höchstleistung betrieben werden, ausgelegt. Diese maximale Leistung beträgt, wie in der Tabelle V.1 zu sehen, 128 Watt. Um noch Ausbaumöglichkeiten zu gewährleisten und uns nicht dem Leistungslimit des Wandlers zu nähern, haben wir uns für einen 48V-12V, 300 Watt DC-DC Wandler von Mean Well² entschieden.

2.1.2 5V Versorgungssystem

Die Leistung des Raspberry Pis ist mit einem Maximum von 6.2 Watt sehr klein und daher ist die Wahl des Spannungswandlers in diesem Fall nicht wirklich davon abhängig. Auch die Komponenten, welche angeschlossen werden, haben grundsätzlich keine erwähnenswerte Wirkleistung und müssen daher nicht genau berechnet werden. Nun entschied nur mehr das Preis-Leistungs-Verhältnis sowie die Ausfallsicherheit des Spannungswandlers die Wahl. Daher haben wir uns für einen 48V-5V, 30 Watt DC-DC Wandler von Meanwell³ entschieden.

2.1.3 Abschalten der Spannungswandler

Das Abschalten der Spannungswandler ist nicht notwendig, da diese - wie schon im Abschnitt 2.1 erklärt - bei einem anliegenden Fehler automatisch abschalten. Ebenso wird beim Abschalten des Motorrades über die BMS jegliches andere Bauteil von der Spannungsversorgung getrennt. Was die Spannungswandler vom Entladen des Akkus abhält.

²Datenblatt: siehe Anhang 0.2

³Datenblatt: siehe Anhang 0.1

3 Steuerung der Peripherie

Die Grundfunktionen wie Beleuchtung, Hupe und Blinker werden hier als Peripherie bezeichnet. Diese sollten so einfach wie möglich und vom Lenker aus zu bedienen sein. Ebenso müssen sie verlässlich gesteuert werden können. Daher haben wir uns entschieden diese Funktionen ebenso über den Raspberry Pi zu steuern, da dieser bei einem Fehler der Motorsteuerung über den eingebauten Puffer gespeist werden kann und daher diese wichtigen Funktionen bis zu einem sicheren Stillstand weiter betrieben und gesteuert werden können.

Dennoch ist in der Plan in Zukunft die Motorsteuerung, welche ebenso in der Lage wäre die Ausgänge abhängig von den Eingängen zu schalten, diese Aufgabe übernehmen zu lassen, solange die Ausfallsicherheit ebenso gegeben wäre. Der Vorteil dieser Methode ist die Schaffung einer Zentralen Steuereinheit, welche alle Steueraufgaben in einem Bauteil vereinen kann.

3.1 Hardware

3.1.1 Input

Man kann einen GPIO Pin entweder als Eingang oder als Ausgang betreiben. Als Eingang kann er die Zustände High und Low einnehmen. Zum Beispiel von einem Schalter oder Taster. In der Regel beschaltet man die GPIOs des Raspberry Pis mit Widerständen, um Eingänge auf einen definierten Pegel zu setzen oder um den Strom zu begrenzen. Standardmäßig werden 10k Widerstände benutzt. Ob Pullup oder Pulldown ist grundsätzlich gleichgültig. Wir benutzen für das Einlesen der Eingänge 10k Pulldown Widerstände, um nicht immer eine Spannung an den Eingängen des Raspberry Pis anliegen zu haben.

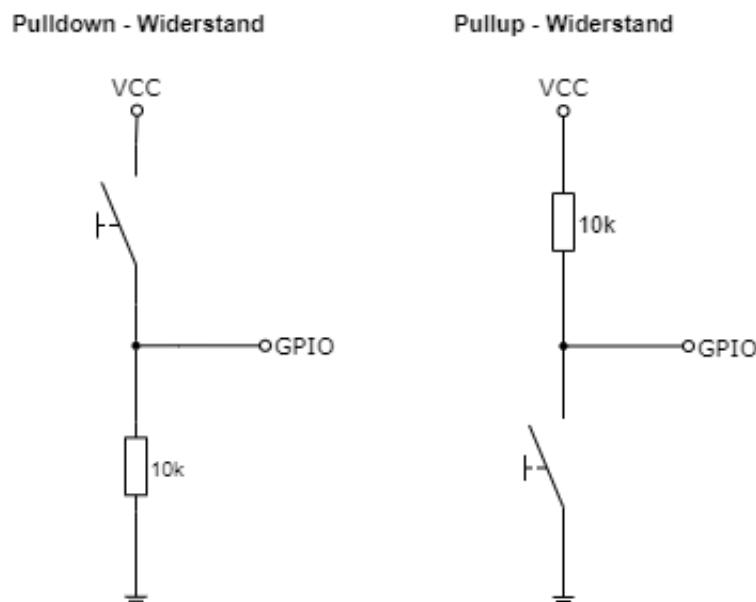


Abbildung V.3: Anschlussplan Eingänge

Pullup Widerstand

Bei dem Nutzen eines Pullup Widerstands wird der GPIO Pin mit einem Widerstand auf die Spannung von VCC gezogen. Der Grundzustand des Eingangs ist dann High. Mit einem Schalter oder Taster wird der Eingang dann gegen Ground gezogen. Das heißt er hat solange der Schalter geschlossen ist, liegt das Massepotential am Eingang an.

Pulldown Widerstand

Bei dem Nutzen eines Pulldown Widerstands wird der GPIO Pin mit einem Widerstand auf die Spannung von Ground gezogen. Der Grundzustand des Eingangs ist dann Low. Mit einem Schalter oder Taster wird der Eingang dann gegen VCC gezogen. Das heißt er hat solange der Schalter geschlossen ist, liegt das Versorgungspotential am Eingang an.

3.1.2 Output

Hierbei werden die GPIOs als Ausgang verwendet. Sie sind verbunden mit den Eingängen eines 4 Channel Relais Moduls, welches über die 5V direkt von dem Raspberry Pi gespeist wird. Hiermit ist es nun möglich die 12V der Peripherie zu schalten und

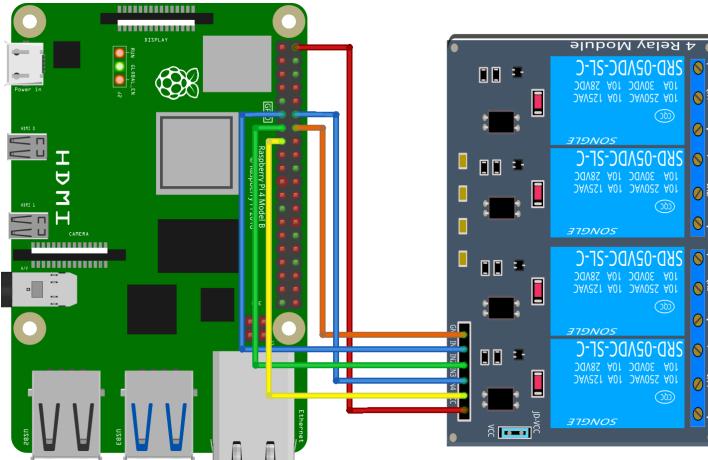


Abbildung V.4: Anschlussplan Relais

3.2 Software

Wichtig bei der Programmierung war in diesem Fall die dauerhafte Verfügbarkeit der Grundfunktionen sowie die einfache Integrierbarkeit von alten und neuen Bauteilen. Dies wurde erreicht durch die Anwendung verschiedener Bibliotheken.

Die Wichtigsten in dieser Anwendung waren:

3.2.1 gpiozero

Eine Bibliothek, welche das einfache und schnelle Integrieren neuer Ein- oder Ausgänge ermöglicht. Dadurch können schnell und einfach Änderungen an dem Steuerverhalten der Peripherie gemacht werden. Hierzu kann die Klasse Button und DigitalOutputDevice verwendet werden. Dieser können Parameter wie die gewünschte Debounce Zeit und verwendete Pull-Widerstände übergeben werden. Es werden nur drei Zeilen Code benötigt um einen Aus- oder Eingang zu definieren und anzusteuern⁴

3.2.2 threading

Eine Bibliothek, welche es ermöglicht, einen eigenen Thread⁵ für das Programm zu öffnen, wodurch die Anwendung ohne Einflüsse oder Unterbrechungen anderer Programme weiter arbeiten kann. Der folgende Programmcode zeigt einen Ausschnitt der Klasse zum Steuern der GPIOs.

```
def start(self):
    self.thread=Thread(target=self.runner)
    self.thread.daemon=True
    self.thread.start()
```

Code Listing V.1: Code zum Starten eines Threads

In diesem Beispiel wird ein Demon Thread erzeugt und gestartet. Das bedeutet dieser Thread wird beim schließen des Programms automatisch mit geschlossen und muss dadurch nicht mehr überwacht werden. Das Target ist die Funktion, welche unabhängig ausgeführt werden soll.

⁴siehe: Anhang

⁵gleichzeitig laufende Aufgabe

4 Benutzeroberfläche

Die Benutzeroberfläche stellt die Verbindung zwischen dem Nutzer und dem Motorrad dar. Sie sollte während der Fahrt die Instrumententafel des Motorrades ersetzen und dem Nutzer die wichtigsten Fahrinformationen anzeigen. Sobald das Motorrad zum Stillstand gekommen ist, wird es möglich Einstellungen zu ändern und die aufgezeichneten Fahrdaten anzeigen zu lassen. Ebenso können der Akkuladestatus und Informationen über Fehler im System entnommen werden.

4.1 Hardware

Zur Anzeige und Bedienung wird ein 11.6 Zoll kapazitives Touch LCD Display verwendet. Es besitzt eine Full HD Auflösung (1920x1080), was für eine professionelle Darstellung essentiell ist. Ebenso hat es ein schützendes ABS Gehäuse, welches trotz fehlender IP Zertifizierung das Abdichten ermöglicht. Die Versorgungsspannung beträgt 12V, was ident zu den anderen Komponenten am Motorrad ist und daher die Versorgung sehr vereinfacht, es kann also über den gleichen Spannungswandler versorgt werden.



Abbildung V.5: Paneel Maße

Die Auflösung und die Größe des Paneels wirkt sich stark auf das Design der Benutzeroberfläche aus. Es muss die Größe der Icons und der anderen Designelemente so angepasst werden, dass sie einerseits gut ersichtlich und andererseits einfach über Berührung zu bedienen sind.

4.1.1 Befestigung

Besser: In das Gehäuse des Paneels sind M4 Verschraubungen in einem Raster von 75mm x 75mm integriert und kann daher einfach an Wänden oder Platten verschraubt werden. Um den Bildschirm nun in einer ähnlichen Position wie die Instrumententafel zu befestigen wurde eine 100mm x 210mm x 1.5mm Aluminium Platte - wie in der Abbildung zu sehen - gebogen und mit Löchern versehen. Um diese Halterung nun an dem Motorrad zu befestigen werden die Verschraubungen der alten Instrumententafel verwendet.



Abbildung V.6: Befestigung des Displays

4.2 Software

Bevor die Software für die Benutzeroberfläche verfasst wurde, mussten das Design, die Funktionen sowie die angezeigten Informationen geplant werden, um einen reibungslosen Arbeitsablauf beim Entwickeln des Frontends zu gewährleisten. Design Elemente wurden zuvor in Adobe Illustrator vorgefertigt. In den folgenden Seiten wird das Ergebnis dieses Prozesses erläutert.

4.2.1 Aufbau

Die nachfolgende Abbildung zeigt den grundsätzlichen Programmaufbau der Benutzeroberfläche. Die einzelnen Fenster werden als Tabelle mit ihren angezeigten Informationen dargestellt. Dies ist wichtig da jede dieser Informationen vom Backend an das Frontend gesendet werden müssen. Ebenso sind in den letzten Zeilen der Tabellen die QML-Elemente zur Navigation zwischen den einzelnen Fenstern niedergeschrieben. Diese müssen auch schon in der frühen Phase der Entwicklung der Benutzeroberfläche definiert werden.

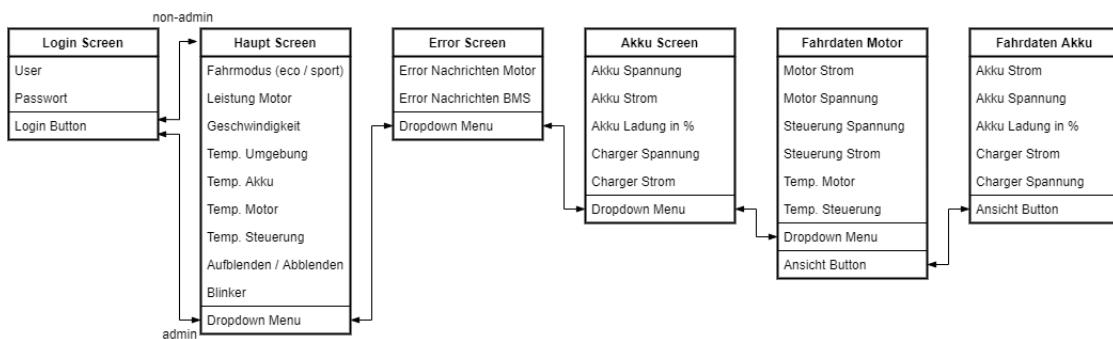


Abbildung V.7: Aufbau der Graphischen Benutzeroberfläche

4.2.2 Nutzer / Berechtigungen

4.3 Komponenten

Komponenten sind wiederverwendbare, gekapselte QML-Elemente mit genau definierten Schnittstellen. Komponenten werden häufig durch Komponenten-Dateien definiert, das heißt durch QML-Dateien. Wichtig ist dabei die Definition von Schnittstellen sowie Properties und Signals.

Properties

Einer Property eines Objektes kann ein statischer Wert zugewiesen werden, der konstant bleibt, bis ihm explizit ein neuer Wert zugewiesen wird. Um QML und seine integrierte Unterstützung für dynamisches Objektverhalten optimal zu nutzen, verwenden die meisten QML-Objekte jedoch Propertybindings.

Dies sind eine Kernfunktion von QML, mit der Beziehungen zwischen verschiedenen Objekteigenschaften festgelegt werden können. Wenn sich die Abhängigkeiten einer Property im Wert ändern, wird die Eigenschaft automatisch gemäß der angegebenen Beziehung aktualisiert. Hinter den Kulissen überwacht die QML-Engine die Abhängigkeiten der Eigenschaft. Wenn eine Änderung erkannt wird, wertet die QML-Engine den Bindungsausdruck erneut aus und wendet das neue Ergebnis auf die Eigenschaft an.

Java-Script-Funktionen

Programmlogik kann auch in Java-Script-Funktionen definiert werden. Diese können in QML-Dokumenten definiert und von Signalhandlern, Eigenschaftsbindungen oder Funktionen in anderen QML-Objekten aufgerufen werden. Solche Methoden werden häufig als Inline-Java-Script-Funktionen bezeichnet, da ihre Implementierung im QML-Dokument statt in einer externen Java-Script-Datei enthalten ist.

4.3.1 Navigationsmenü

Das Navigations-Menü ist ein Dropdown-Menü, welches zur Navigation zwischen den verschiedenen Fenstern benutzt wird. Sobald man sich eingeloggt hat wird das Menü angezeigt und die einzelnen Untermenüs können aufgerufen werden. Das Menü wird abhängig von den Berechtigungen des Benutzers angepasst.



Abbildung V.8: GUI Komponente - Navigation Menü

Buttons

Die Navigation wird über das QML-Element *Mousearea*, welche direkt über den Icons der einzelnen Navigationselemente platziert wurde, gesteuert. Nun kann mit dem Befehl *onClicked* eine Funktion aufgerufen werden, welche das gewünschte Fenster sichtbar macht, sowie das Navigationsmenü wieder nach oben fahren lässt.

In dieser Funktion wird ebenso die Berechtigung des Nutzers über eine Globale Variable, welche beim Anmelden durch ein Signal gesetzt wird, abgefragt. Falls die Berechtigung die ausgewählte Funktion nicht zulässt wird ein Informationstext ausgegeben und das Menü wiederum geschlossen.

Abmelden

Wird der Abmeldebutton gedrückt, werden die Anmeldeinformationen zurückgesetzt und dem Nutzer wird wieder das Anmeldefenster angezeigt, wo er sich nun mit anderen Anmelde-Informationen einloggen kann.

4.3.2 Balken Anzeige

Die Komponente Balken Anzeige wird in der Benutzeroberfläche zur Visualisierung verschiedener Daten verwendet. Mit ihr können diese übersichtlicher dargestellt werden. Diese Komponente wird in mehreren QML-Dateien verwendet, daher sind Properties zur Anpassung notwendig. Die wichtigsten davon sind:

- Wert - Der aktuelle Wert, welcher am Balken angezeigt werden sollte.
- Anfangswinkel - Der Winkel an dem der Balken entspringt.
- maximaler Wert - Der maximal zu erreichende Wert. Dieser bestimmt die Länge des Hintergrundbalkens
- Hintergrundfarbe - Farbe des Hintergrundbalkens (in der Abbildung grau)
- Balkenfarbe - Farbe des Anzeigebalkens (in der Abbildung blau)

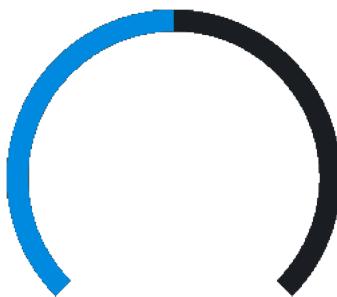


Abbildung V.9: GUI Komponente - Balken Anzeige

Über ein *Signal* kann nun über das Backend der Wert des Balkens verändert werden und in Echtzeit angezeigt werden.

4.3.3 Modus Anzeige

Diese Komponente befindet sich auf der Instrumententafel. Der Modus kann über einen Button am Lenker, welcher mit der Curtis Steuerung verbunden ist, geändert werden. Sie dient zum Anzeigen der derzeitig gewählten Fahrmodi und wird über ein *Signal* aus dem Backend gesteuert, welches mit der Listener Klasse verbunden ist.



Abbildung V.10: GUI Komponente - Modus Anzeige

Die Modus Anzeige ist eine einfache Komponente. Sie ist ein Item QML-Typ, dadurch wird das Nutzen von *States* möglich. Durch diese können verschiedene Eigenschaften gespeichert und über einen kurzen Befehl wiederhergestellt werden.

In diesem Fall wird der Punkt ausgefüllt und die Farbe des Textes geändert.

4.3.4 Graph

Die Graph Komponente befindet sich auf dem Diagnose Fenster und dient zum Anzeigen der in der Datenbank gespeicherten Fahrdaten. Sie verfügt über einen Ladebalken, welcher nach dem Auswählen der anzuseigenden Daten den Fortschritt des Auslesens der Datenbank anzeigt. Dieses Auslesen läuft über eine Funktion der Fetcher Klasse.

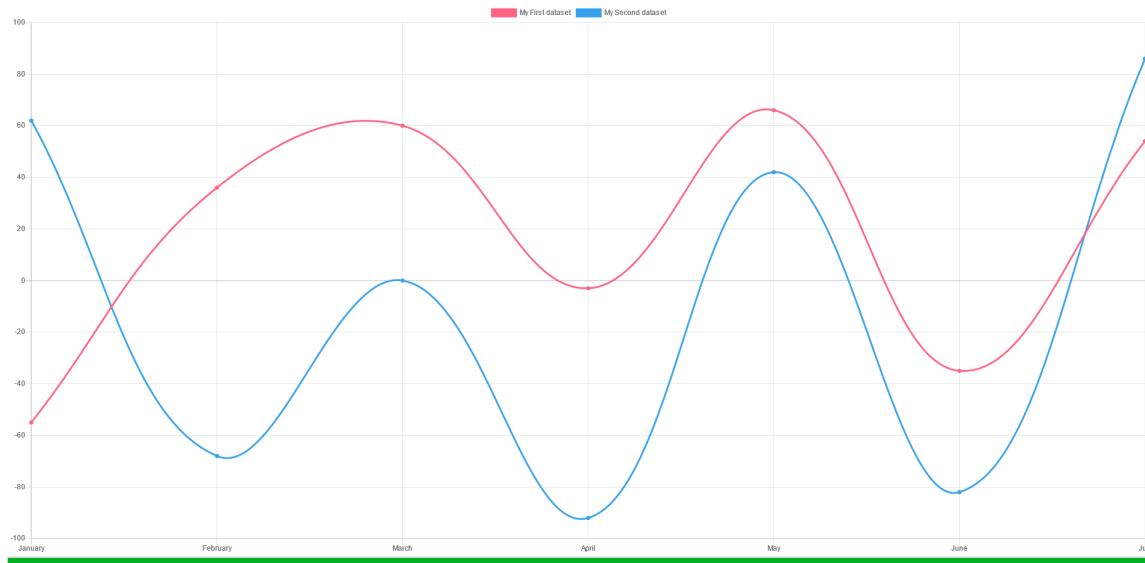


Abbildung V.11: GUI Komponente - Graph

Realisiert wurde diese Komponente über öffentliche Bibliothek *Charts.js* (Referenz), welche durch die Java Script Unterstützung von *QT*, mit der Installation eines Adapters mit dem QML Programm kompatibel ist. Dadurch wird nun das Zeichnen von verschiedenen Graphen ermöglicht.

4.3.5 Weitere Komponenten

Wie schon bei der Fahr Modi Komponente, wurden weitere Objekte, welche zwischen zwei Zuständen hin und her gestalten werden müssen oder eine bestimmte Handlung ausführen sollten, über eine eigene QML Komponente verwirklicht. Diese ermöglicht nun wiederum das wiederholte verwenden dieser Komponente, sowie das Ansteuern über *States* und *Signale*. Diese Komponenten sind:

Umschaltkomponenten

Dies sind Komponenten, welche über ein *Signal* angesteuert werden und dann mithilfe von *States* ihr Aussehen verändern.

- Blinker Rechts
- Blinker Links
- Tagfahrlicht
- Fernlicht

Touch Komponenten

Dies sind Komponenten, welche eine *Mousearea* mit einer Graphik verbindet. Wird diese Komponente nun mit einem Mausklick ausgewählt, kann eine Funktion ausgeführt oder ein *Signal* ausgesendet werden.

- Abmeldebutton
- Navigationsmenübuttons
- Menü-Öffnen-Button

4.4 Programm Fenster

4.4.1 Login

Das Login Fenster dient zur Autorisierung des Benutzers. Über die User Kombobox kann der gewünschte Nutzer mit der dazugehörigen Berechtigung ausgewählt werden (siehe Abschnitt). Das Fenster dient ebenso zur Darstellung der Logos unserer Sponsoren



Abbildung V.12: GUI Fenster - Login Menü

Um sich einzuloggen muss nur mehr das Passwort im *Textfeld* darunter eingegeben werden. Nach dem Drücken des Login Pfeiles (Neben dem Passwort Textfeld) werden die Login Daten an das Backend versendet. Diese vergleicht die Daten über die Datenbankschnittstelle⁶ und loggt, insofern das richtige Passwort gegeben ist, den Benutzer ein. Wird jedoch das Passwort falsch eingegeben, wird eine rote Fehlermeldung angezeigt.

4.4.2 Fahrdaten

Dieses Fenster dient als Ersatz für die Instrumententafel des Motorrades. Es ist daher das einzige während der Fahrt ersichtliche Fenster. Es Zeigt alle wichtigen Fahrdaten wie Leistung, Geschwindigkeit und Akkuladestand, sowie den Aktuellen Fahrmodus an. Diese Anzeige kann jedem Nutzer, unabhängig der Berechtigung, angezeigt werden.

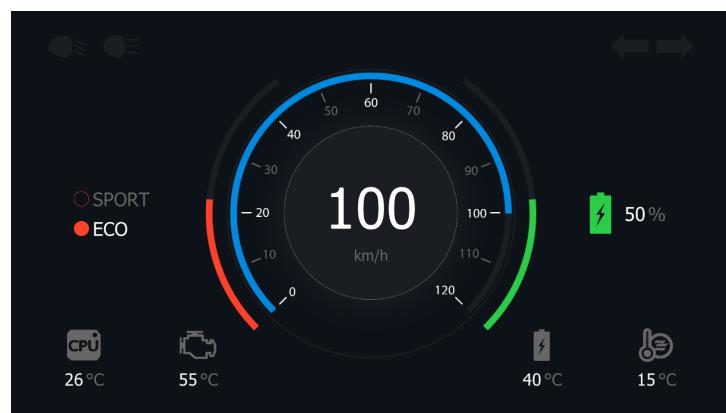


Abbildung V.13: GUI Fenster - Fahrdaten

Der Status der Blinker und der Beleuchtung wird ebenso wie die Temperaturen der verschiedenen Komponenten des Motorrades angezeigt. Derzeit ist diese Ansicht starr und kann noch nicht geändert werden. Doch es ist geplant in Zukunft weitere Designs, welche in den Einstellungen geändert werden können, zu implementieren.

⁶siehe Abschnitt

4.4.3 Akku- und Ladedaten

Dieses Fenster dient Ähnlich wie das der Fahrdaten zur Darstellung wichtiger Daten. Es können alle wichtigen Daten bezüglich Akku und BMS auf einem Blick abgelesen werden.

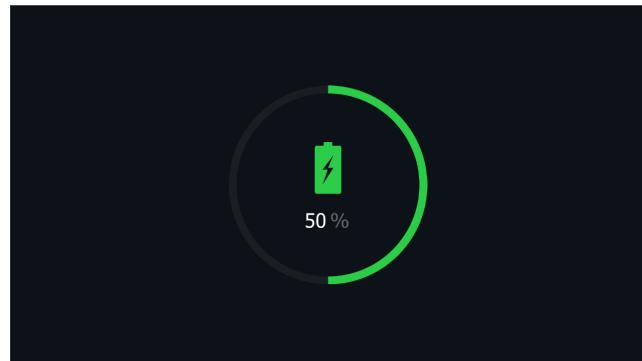


Abbildung V.14: GUI Fenster - Akkudaten

Dieses Fenster wird Standardmäßig während dem Laden des Fahrzeuges angezeigt. Es benötigt also keine bestimmte Berechtigung und kann von jedem Nutzer über das Navigations-Menü angezeigt werden.

4.4.4 Fahrdaten Diagnose

Hier können die Fahrdaten, welche während der Fahrt dauerhaft von der Motorsteuerung versendet und vom Raspberry Pi verarbeitet und in einer Datenbank gespeichert werden, in Graphen angezeigt werden. Hierzu wird die *Graph Komponente*⁷ verwendet.

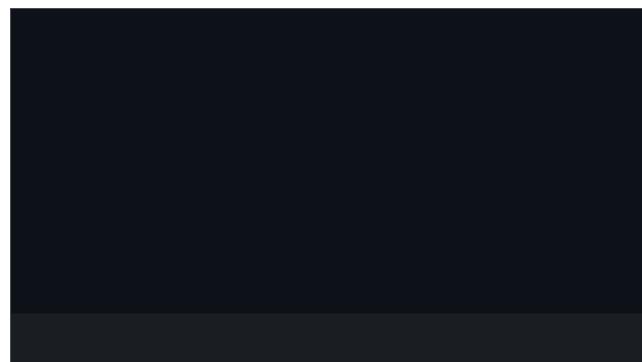


Abbildung V.15: GUI Fenster - Fahrdaten Diagnose

Diese wird, um während der Fahrt die volle Prozessorleistung zu gewährleisten, erst auf Knopfdruck geladen. In der Menüleiste am unteren Ende des Fensters können verschiedene Vorlagen ausgewählt werden, welche über fixe Datensätze verfügen. Das bedeutet es werden erst beim Auswählen des anzuseigenden Datensatzes die Daten der letzten Stunde ausgelesen und in den Graphen geladen.

⁷siehe Abschnitt 4.3.4

4.4.5 Fehler

Sobald ein Fehlerdatenpaket über den CAN-Bus versendet wird, gibt der Listener Klasse die Fehlercodes an die Bridge Klasse weiter, um sie in diesem Fenster anzuzeigen. In einer *Listview* werden nun die Aktiven Fehler angezeigt. Solange Fehler anliegen werden diese angezeigt und erst sobald die zuvor angezeigten Fehler nicht mehr anliegen werden diese aus der Liste gelöscht.

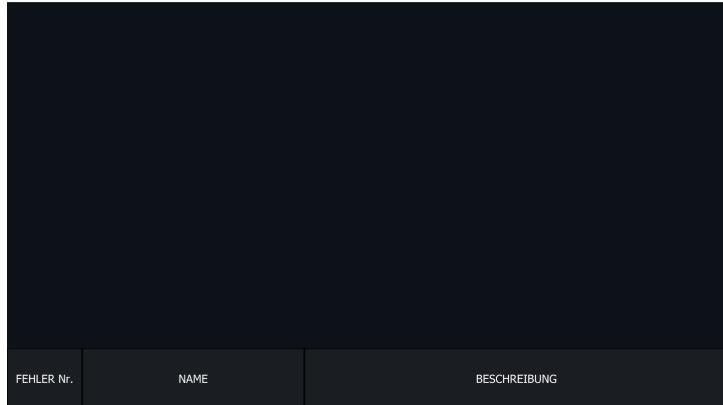


Abbildung V.16: GUI Fenster - Fehler Liste

Der Name und die Beschreibung der Fehler, wird über eine Fehlertabelle⁸ in der Datenbank ausgelesen. Diese werden während dem Testen des Motorrades gesammelt und häufig auftretende Fehler werden in diese Tabelle eingetragen.

4.4.6 Nutzer und Berechtigungen

Um sicherzustellen das das Nutzen der Benutzeroberfläche nicht für jeden uneingeschränkt möglich ist wurde ein Anmeldesystem mit drei verschiedenen Benutzern, welche jeweils verschiedene Berechtigungen besitzen, umgesetzt.

ADMIN - Passwort: 53AC2

Dieser Benutzer verfügt über uneingeschränkte Rechte und kann auf jede verfügbare Funktion sowie Funktionen, welche noch in Entwicklung sind zugreifen.

USER - Passwort: 5AHET

Dieser Benutzer verfügt über wenig eingeschränkte Rechte. Er kann auf jede fertig entwickelte Funktion zugreifen kann jedoch keine Änderungen an der Benutzeroberfläche vornehmen und hat ebenso keinen Zugriff auf experimentelle Funktionen.

GUEST - Passwort: 00000

Dieser Benutzer verfügt nur über sehr eingeschränkte Rechte. Er wird zum Präsentieren der Maschine benutzt und lässt den unerfahrenen Nutzer nur auf die Fahrdaten und Akku- und Ladedaten zugreifen. Ebenso sollte in Zukunft die Leistung des Motorrades beim einloggen dieses Benutzers stark eingeschränkt werden, um Unfälle zu verhindern.

⁸siehe Abschnitt

4.5 Realisierung der Benutzeroberfächer

4.5.1 QML

QML⁹ ist eine deklarative Sprache, mit der Benutzeroberflächen anhand ihrer visuellen Komponenten und ihrer Interaktion und Beziehung zueinander beschrieben werden können. Es ist eine gut lesbare Sprache, die entwickelt wurde, um die dynamische Verbindung von Komponenten zu ermöglichen und die einfache Wiederverwendung und Anpassung von Komponenten innerhalb einer Benutzeroberfläche erlaubt. Es bietet Syntax mit Unterstützung für Java-Script-Ausdrücke in Kombination mit dynamischen Eigenschaftsverbindungen.

4.5.2 Qt-Quick

Das Qt-Quick-Modul ist die Standardbibliothek zum schreiben von QML-Anwendungen. Während das QML-Modul die Engine und die Sprachinfrastruktur bereitstellt, bietet das Qt Quick-Modul alle grundlegenden Typen, die zum Erstellen von Benutzeroberflächen mit QML erforderlich sind. Es bietet eine visuelle Zeichenfläche und Typen zum Erstellen und Animieren visueller Komponenten, zum Empfangen von Benutzereingaben, zum Erstellen von Datenmodellen und Ansichten sowie zum verzögerten Objektinstanziieren. Es können problemlos flüssige, animierte Benutzeroberflächen in QML erstellt werden. Diese Benutzeroberflächen können mit beliebigen Backend Bibliotheken verbunden werden.

4.5.3 Slots und Signals

Slots und Signals werden in QML zur ereignisgesteuerten Kommunikation zwischen Frontend und Backend verwendet. In der folgenden Illustration wird diese anhand eines einfachen Beispiels erklärt.

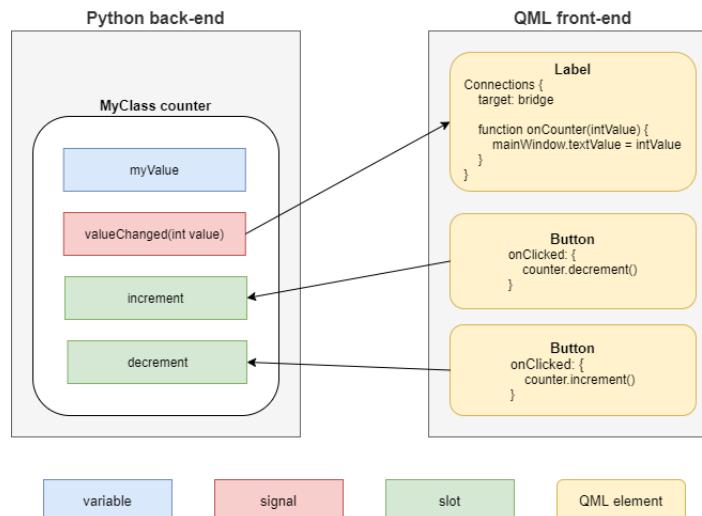


Abbildung V.17: Slots und Signals Konzept

Signale

Diese können als Mitteilungen angesehen werden, welche über das Aufrufen der `signal.emit()` Funktion vom Backend an das Frontend gesendet wird. Im Frontend wird wiederum eine eigens definierte Funktion benötigt um dem Wert einem Property eines QML Elements zuzuweisen.

Slots

Slots sind Call-Back Funktionen, welche im Backend definiert werden und sind über die Bridge Klasse mit dem Frontend verknüpft. Dadurch können diese Funktionen im Frontend aufgerufen und mit Signalen verbunden werden. Sie stellen daher die wichtigste Verbindung zwischen dem Programm und der Benutzeroberfläche dar.

⁹Qt Modeling Language

4.5.4 Bridge

Die Bridge Klasse wird für die Kommunikation zwischen den einzelnen Modulen des Backends mit denen der graphischen Benutzeroberfläche.

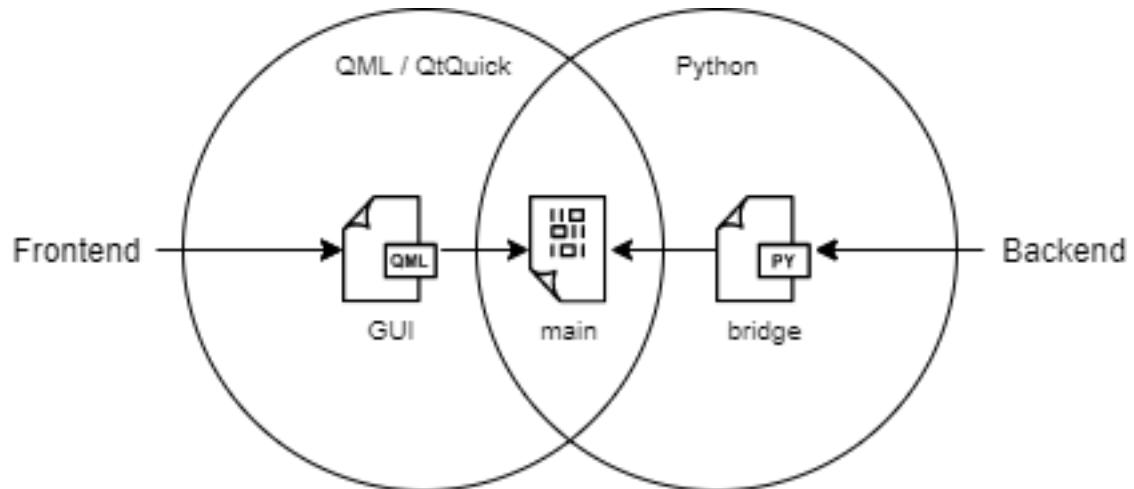


Abbildung V.18: Verbindung Frontend zu Backend

5 Kommunikation

Um Daten zwischen den mehreren Steuereinheiten des Motorrades zu versenden, muss eine Echtzeit-Kommunikation über ein Bussystem gewährleistet werden. Die Entscheidung ist auf das Controller Area Network Bussystem (CAN-Bus) gefallen. Ausschlaggebend für diese Entscheidung war der Curtis Motorcontroller, dieser verfügt über eine serielle Schnittstelle (RS-232) sowie ein CAN-System. Für unserer Anwendung bietet das CAN-System eine größere Ausbaufähigkeit sowie größere Übertragungsraten, weshalb wir uns letztendlich auch dafür entschieden haben.

5.1 Hardware

5.1.1 CAN-Modul

Da der Raspberry Pi selbst nicht über ein CAN-System verfügt, erfolgt der Anschluss an die Busleitung über ein externes CAN-Modul, welches über das Serial Peripheral Interface (SPI) mit dem Raspberry Pi kommuniziert, welches wie folgt angeschlossen werden muss:

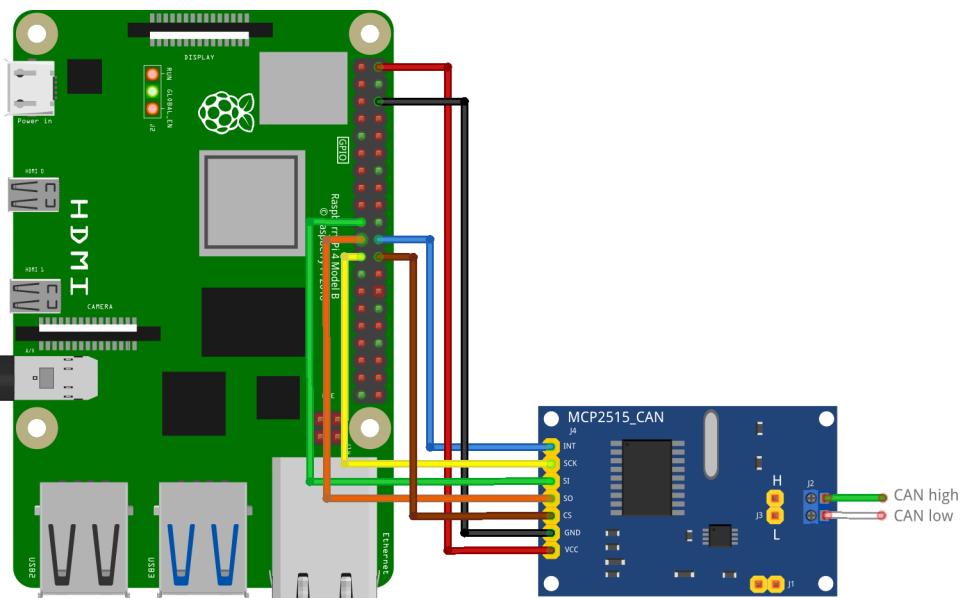


Abbildung V.19: Anschlussplan CAN-Modul

Die Kommunikation wird über zwei Komponenten ermöglicht. Einen MCP2562 Transceiver, welcher für die Verarbeitung der Nachrichten zuständig ist und ein MCP2515 CAN Interface, welches die Daten zwischenspeichert und sich um das Versenden der Nachrichten kümmert. Gemeinsam mit dem Mikrocomputer ergibt dies nun einen CAN Node, welcher fähig ist Nachrichten zu versenden und zu empfangen.

5.1.2 Netzwerkstruktur

Ein CAN-Netzwerk wird standardmäßig als Bus- oder Sternkopplung aufgebaut. Wir haben uns bewusst für die Bustopologie entschieden, da Sternkopplungen nur in bestimmten Anwendungen Gebrauch finden und noch dazu markante Nachteile besitzt.

Es müsste zum Beispiel eine zentrale Steuereinheit den Nachrichtenverkehr steuern, ebenso gibt uns die niedrige Anzahl an Teilnehmern im Netzwerk nicht einmal die Möglichkeit ein anderes System zu verwenden. Erst wenn wir in Zukunft das CAN-Netzwerk um Sensoren und Aktoren erweitern würden, müsste weitere Zeit in die Planung des Netzwerks investiert werden.

5.2 Listener

Die Listener Klasse ist dafür zuständig den Datenverkehr am Bus zu überwachen und geordnet an die Datenbankschnittstelle (Fetcher) sowie die Schnittstelle zum Frontend (Bridge) weiterzugeben.

5.2.1 Konfigurieren der Schnittstelle

```
def start(self):
    self.thread=Thread(target=self.runner)
    self.thread.daemon=True
    self.thread.start()
```

Code Listing V.2: Konfigurieren des CAN Adapters

5.2.2 Empfangen der Daten

6 Fahrdatenspeicher

Das Motorrad sollte seine Fahrdaten abspeichern und darstellen können, um eine Diagnose des gesamten Aufbaus zu gewährleisten. Dafür wird eine Datenbank verwendet, nämlich die weit verbreitete Software mySQL verwendet.

6.1 Datenbankstruktur

6.1.1 Benutzer System

users		
Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
username	varchar(50)	Name des Benutzers
password	varchar(255)	Eingegebenes Passwort

Tabelle V.2: Datenbankstruktur der Benutzer Tabelle

6.1.2 Motor Daten

Datenpacket 1

data1		
Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
current_rms	int	Ausgangsstrom der Motorsteuerung
controller_temp	int	Temperatur des Kontrollers
motor_temp	int	Temperatur des Motors
motor_torque	int	Drehmoment am Motor
modulation_depth	int	Modulationsgrad des Kontrollers

Tabelle V.3: Datenbankstruktur der Datenpaket 1 Tabelle

id und enabled sind nicht signiert, wobei id der Primärschlüssel mit Auto-Inkrement ist. step_id ist ein Zeilerattribut, das für die Reihenfolge der Schritte verwendet wird, wie sie in der grafischen Oberfläche angezeigt werden. Der kaskadierte Fremdschlüssel tutorial_id wird für die Zuordnung zu dem jeweiligen Tutorial verwendet. FÜR die

Datenpacket 2

data2		
Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
id_data_1	int	Fremdschlüssel von Datenpacket 1
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
capacitator_volts	int	Eingangsspannung der Motorsteuerung
bdi_percentage	int	
interlock	int	
throttle_command	int	Stellung des Gasgriffes
brake_command	int	Stellung der Motorbremse (software)
emr_state	int	

Tabelle V.4: Datenbankstruktur der Datenpaket 2 Tabelle

Datenpacket 3

data3		
Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
id_data_2	int	Fremdschlüssel von Datenpacket 2
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
vehicle_acceleration	int	Eingangsspannung der Motorsteuerung
vehicle_odometer	int	Vom Motorrad zurückgelegte Strecke
time_to_capture_speed	int	Zeit bis zur eingestellten Geschwindigkeit
time_to_capture_distance	int	Zeit bis zur eingestellten Distanz
braking_distance_captured	int	Zeit von Bremsbeginn bis zu Stillstand
distance_since_stop	int	Zurückgelegte Distanz seit dem Stillstand

Tabelle V.5: Datenbankstruktur der Datenpaket 3 Tabelle

Datenpacket 4 - Fehler

data4		
Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
id_data_3	int	Fremdschlüssel von Datenpacket 3
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
vehicle_acceleration	int	Eingangsspannung der Motorsteuerung
vehicle_odometer	int	Vom Motorrad zurückgelegte Strecke
time_to_capture_speed	int	Zeit bis zur eingestellten Geschwindigkeit
time_to_capture_distance	int	Zeit bis zur eingestellten Distanz
braking_distance_captured	int	Zeit von Bremsbeginn bis zu Stillstand
distance_since_stop	int	Zurückgelegte Distanz seit dem Stillstand

Tabelle V.6: Datenbankstruktur der Fehler-Datenpaket Tabelle

6.1.3 Fehler Tabelle

Diese Tabelle ist ebenso in der Datenbank gespeichert. Sie beinhaltet die Fehlercodes der Motorsteuerung mit weiteren Informationen wie Name und Beschreibung. Sie muss per Hand angelegt werden und sollte während dem Testen der Maschine mit dokumentiert, um sie dann in das Fehler Fenster der Benutzeroberfläche zu implementieren.

6.1.4 Akku Daten

Da die Kommunikation der BMS noch nicht umgesetzt wurde, werden von Ihr keine Daten auf den CAN-Bus gelegt und daher muss auch keine Tabelle dafür erstellt werden. Somit werden alle Daten über Akku und Ladestand über die Motorsteuerung empfangen.

6.2 Handler

Die Handler Klasse hat die Aufgabe eine Verbindung zwischen der Datenbank und dem Programm herzustellen. Dazu verwendet sie die MySQL connector Bibliothek. Mit dieser können SQL Skripte direkt in die Python Klasse integriert werden ohne dass externe SQL Skript Dateien geöffnet werden müssen.

6.2.1 Konfigurieren der Schnittstelle

```
# Database credentials
dbhost = "localhost"
dbuser = "root"
dbpass = "8Yf!97mmnZbvYJe"
dbname = "login_data"

# Connection to the database
con = mysql.connector.connect(host=dbhost, user=dbuser, password=dbpass,
                               database=dbname)

# Creating cursor for database queries
cursor = con.cursor()
```

Code Listing V.3: Konfiguration der Datenbankschnittstelle

6.2.2 SELECT Befehl

Der SELECT Befehl wird zum Abrufen von Daten einer Tabelle verwendet und besitzt mehrere Parameter.

```
INSERT INTO `users` ('id', 'username', 'password')
VALUES (1, 'USER', '5AHET');
```

```
INSERT INTO `users` ('id', 'username', 'password')
VALUES (2, 'GUEST', '00000');
```

```
INSERT INTO `users` ('id', 'username', 'password')
VALUES (3, 'ADMIN', '53AC2');
```

Code Listing V.4: Konfiguration der Datenbankschnittstelle

Der Befehl kann in drei Bereiche unterteilt werden. In der ersten Zeile werden die Attribute, die von der Tabelle Tutorials ausgegeben werden sollen, definiert. Da sich diese jedoch nicht in Tutorials befinden, werden im mittleren Abschnitt alle Tabellen durch den Befehl JOIN anhand der Fremdschlüssel verbunden. Die anzugebenden Ergebnisse werden durch den *WHERE* Befehl gefiltert werden. Dieser Befehl zeigt die Attribute id, step id sowie detection aller Tutorials vom Programm mit dem Namen Windows Settings an.

6.2.3 INSERT Befehl

Der INSERT Befehl wird zum Schreiben von Daten in eine Tabelle verwendet.

```
INSERT INTO `users` ('id', 'username', 'password')
VALUES (1, 'USER', '5AHET');
```

```
INSERT INTO `users` ('id', 'username', 'password')
VALUES (2, 'GUEST', '00000');
```

```
INSERT INTO `users` ('id', 'username', 'password')
VALUES (3, 'ADMIN', '53AC2');
```

Code Listing V.5: Einfügen der Benutzer und Passwörter über den

Dieser Befehl kann in zwei Teile unterteilt werden. In der ersten Zeile wird die Zieltabelle mit den Attributen angegeben. In der zweiten Zeile folgen die Werte, die in der Datenbank erfasst werden sollen. Hierbei ist die Reihenfolge der Attribute zu beachten. Das Attribut id wird standardmäßig von dem Datenbanksystem vergeben, es ist daher möglich das Attribut im Befehl zu vernachlässigen. Ebenso unterstützt MySQL als Wert die Funktion *NOW()*, wodurch intern der aktuelle Zeitstempel abgespeichert wird. Dadurch kümmert sich das Datenbanksystem um die korrekten Werte für das *date_time* Attribut.

Kapitel VI

Antriebsstrang

Kapitel VII

Akku und Ladekonzept

Kapitel VIII

Endergebnis

Anhang A

Arbeitsnachweis

1 Zeitplan

2 Kosten

Anhang B

Programmcode

Anhang C

CAD-Zeichnungen

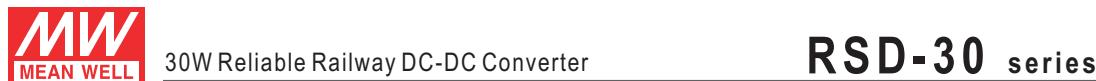
Anhang D

Schaltpläne

Anhang E

Datenblätter

0.1 Mean Well RSD-30H-5



■ Features

- Compliance to EN50155 and EN45545-2 railway standard
- Ultra compact and 1U low profile(25mm)
- 4:1 wide input range
- No minimum load required
- Protections: Short circuit / Overload / Over voltage / Input reverse polarity
- 4000VDC I/O isolation (reinforced isolation)
- Half encapsulated , cooling by free air convection
- -40~+70°C wide working temperature
- Built-in constant current limiting circuit
- LED indicator for power on
- 3 years warranty

■ Applications

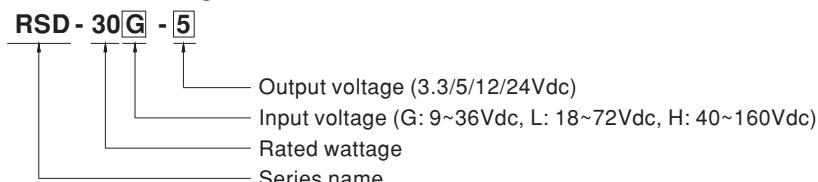
- Bus,tram,metro or railway system
- Wireless network
- Telecom or datacom system
- Highly vibrating, highly dusty, extremely low or high temperature harsh environment

■ Description

RSD-30 is a 30W enclosed type DC-DC reliable railway converter. This series is compliant with EN50155/IEC60571 railway standard, constituting three types of models with 4:1 wide but different input ranges 9~36V/18~72V/40~160V, suitable for railway and all kinds of transportation systems exploiting the frequently used standard input voltages such as 12V, 24V, 36V, 48V, 72V, 96V and 110V. Various output voltages, 3.3V, 5V, 12V and 24V are available for selection.

This series has the capability of working under -40~+70°C, low ripple and noise, supreme EMC characteristics, 4KVDC I/P-OP, low enclosure profile 25mm and an interior with semi-potted silicone. It does not only well fits the in-car systems or the facilities by rails for railway, trams and buses but also can be used in the harsh environment with high vibration, high dust, extremely low or high temperature, etc.

■ Model Encoding



File Name:RSD-30-SPEC 2017-02-22



30W Reliable Railway DC-DC Converter

RSD-30 series**SPECIFICATION**

MODEL	RSD-30G-3.3	RSD-30G-5	RSD-30G-12	RSD-30G-24	RSD-30L-3.3	RSD-30L-5	RSD-30L-12	RSD-30L-24							
OUTPUT	DC VOLTAGE	3.3V	5V	12V	24V	3.3V	5V	12V	24V						
	RATED CURRENT	6A	6A	2.5A	1.25A	6A	6A	2.5A	1.25A						
	CURRENT RANGE	0 ~ 6A	0 ~ 6A	0 ~ 2.5A	0 ~ 1.25A	0 ~ 6A	0 ~ 6A	0 ~ 2.5A	0 ~ 1.25A						
	RATED POWER	19.8W	30W	30W	30W	19.8W	30W	30W	30W						
	RIPPLE & NOISE (max.) Note.2	70mVp-p	70mVp-p	60mVp-p	50mVp-p	70mVp-p	70mVp-p	60mVp-p	50mVp-p						
	VOLTAGE TOLERANCE Note.3	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%						
	LINE REGULATION	±0.5%	±0.5%	±0.3%	±0.2%	±0.5%	±0.5%	±0.3%	±0.2%						
	LOAD REGULATION	±0.5%	±0.5%	±0.3%	±0.2%	±0.5%	±0.5%	±0.3%	±0.2%						
	SETUP, RISE TIME	120ms, 85ms at full load													
HOLD UP TIME (Typ.)		G type comply with S1 level(3ms) @full load, S2 level(10ms) @80% load; L type comply with S2 level(10ms) @full load													
INPUT	VOLTAGE RANGE CONTINUOUS	9 ~ 36VDC				18 ~ 72VDC									
	EFFICIENCY (Typ.)	84%	85%	86.5%	89%	84%	86%	90%	91%						
	DC CURRENT (Typ.)	1.1A/24V				0.52A/48V									
	INRUSH CURRENT (Typ.)	20A/24VDC				20A/48VDC									
PROTECTION	OVERLOAD	105 ~ 135% rated output power													
		Protection type : Constant current limiting, recovers automatically after fault condition is removed													
	OVER VOLTAGE	3.8 ~ 4.5V	5.75 ~ 7V	13.8 ~ 16.2V	27.6 ~ 32.4V	3.8 ~ 4.5V	5.75 ~ 7V	13.8 ~ 16.2V	27.6 ~ 32.4V						
ENVIRONMENT	WORKING TEMP.	-40 ~ +55°C (no derating); +70°C @ 60% load by free air convection; +70°C (no derating with external base plate)													
	WORKING HUMIDITY	5 ~ 95% RH non-condensing													
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH non-condensing													
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)													
	VIBRATION	10 ~ 500Hz, 5G 10min./1cycle, 60min. each along X, Y, Z axes; Mounting : compliance to IEC61373													
SAFETY & EMC (Note 4)	SAFETY STANDARDS	Meet IEC60950-1 (LVD)													
	WITHSTAND VOLTAGE	I/P-O/P:4KVDC I/P-FG:2.5KVDC O/P-FG:2.5KVDC													
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH													
	EMC EMISSION	Parameter	Standard	Test Level / Note											
		Conducted	EN55032	Class A											
		Radiated	EN55032	Class B											
		Harmonic Current	EN6100-3-2	Class A											
	EMC IMMUNITY	Voltage Flicker	EN6100-3-3	-----											
		Parameter	Standard	Test Level / Note											
		ESD	EN61000-4-2	Level 3, ±8KV air; Level 3, ±6KV contact											
		Radiated Field	EN61000-4-3	Level X											
		EFT / Burst	EN61000-4-4	Level 3, 2KV at power											
		Surge	EN61000-4-5	Level 3, 1KV Line-Line, Level 3, 2KV Line-Earth											
OTHERS	Conducted	EN61000-4-6	Level 3												
	RAILWAY STANDARD	Compliance to EN45545-2 for fire protection; Meet EN50155 / IEC60571 including IEC61373 for shock & vibration, EN50121-3-2 for EMC													
NOTE	MTBF	396.9K hrs min. MIL-HDBK-217F (25°C)													
	DIMENSION	113*60*25mm (L*W*H)													
	PACKING	0.25Kg; 56pcs/15Kg/0.83CUFT													
	1. All parameters NOT specially mentioned are measured at 24.48VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. Strongly recommended that external output capacitance should not exceed 5000uF.														



30W Reliable Railway DC-DC Converter

RSD-30 series**SPECIFICATION**

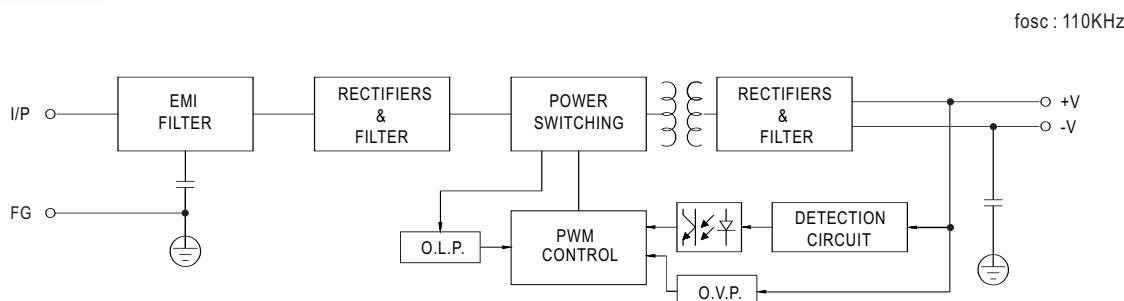
MODEL	RSD-30H-3.3	RSD-30H-5	RSD-30H-12	RSD-30H-24
OUTPUT	DC VOLTAGE	3.3V	5V	12V
	RATED CURRENT	6A	6A	2.5A
	CURRENT RANGE	0 ~ 6A	0 ~ 6A	0 ~ 2.5A
	RATED POWER	19.8W	30W	30W
	RIPPLE & NOISE (max.) Note.2	70mVp-p	70mVp-p	60mVp-p
	VOLTAGE TOLERANCE Note.3	±2.0%	±2.0%	±2.0%
	LINE REGULATION	±0.5%	±0.5%	±0.3%
	LOAD REGULATION	±0.5%	±0.5%	±0.3%
	SETUP, RISE TIME	120ms, 85ms at full load		
	HOLD UP TIME (Typ.)	H-type comply with S2 level(10ms) @ full load		
INPUT	VOLTAGE RANGE CONTINUOUS	40 ~ 160VDC		
	EFFICIENCY (Typ.)	87%	89%	89%
	DC CURRENT (Typ.)	0.23A/110V	0.35A/110V	
	INRUSH CURRENT (Typ.)	20A/110VDC		
PROTECTION	OVERLOAD	105 ~ 135% rated output power Protection type : Constant current limiting, recovers automatically after fault condition is removed		
		3.8 ~ 4.5V	5.75 ~ 7V	13.8 ~ 16.2V
	OVER VOLTAGE	27.6 ~ 32.4V Protection type : Shut down o/p voltage, re-power on to recover		
ENVIRONMENT	WORKING TEMP.	-40 ~ +55°C (no derating) ; +70°C @ 60% load by free air convection ; +70°C (no derating with external base plate)		
	WORKING HUMIDITY	5 ~ 95% RH non-condensing		
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH non-condensing		
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)		
	VIBRATION	10 ~ 500Hz, 5G 10min./1cycle, 60min. each along X, Y, Z axes ; Mounting : compliance to IEC61373		
SAFETY & EMC (Note 4)	SAFETY STANDARDS	Meet IEC60950-1 (LVD)		
	WITHSTAND VOLTAGE	I/P-O/P:4KVDC I/P-FG:2.5KVDC O/P-FG:2.5KVDC		
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH		
	EMC EMISSION	Parameter	Standard	Test Level / Note
		Conducted	EN55032	Class A
		Radiated	EN55032	Class B
		Harmonic Current	EN6100-3-2	Class A
		Voltage Flicker	EN6100-3-3	-----
	EMC IMMUNITY	Parameter	Standard	Test Level / Note
		ESD	EN61000-4-2	Level 3, ±8KV air ; Level 3, ±6KV contact
		Radiated Field	EN61000-4-3	Level X
		EFT / Burst	EN61000-4-4	Level 3, 2KV at power Level 4, 2KV at signal
		Surge	EN61000-4-5	Level 3, 1KV Line-Line, Level 3, 2KV Line-Earth
		Conducted	EN61000-4-6	Level 3
	RAILWAY STANDARD	Compliance to EN45545-2 for fire protection ; Meet EN50155 / IEC60571 including IEC61373 for shock & vibration, EN50121-3-2 for EMC		
OTHERS	MTBF	396.9K hrs min. MIL-HDBK-217F (25°C)		
	DIMENSION	113*60*25mm (L*W*H)		
	PACKING	0.25Kg; 56pcs/15Kg/0.83CUFT		
NOTE	1. All parameters NOT specially mentioned are measured at 110VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. Strongly recommended that external output capacitance should not exceed 5000uF.			



30W Reliable Railway DC-DC Converter

RSD-30 series

■ Block Diagram



■ Input Fuse

There is one fuse connected in series to the positive input line, which is used to protect against abnormal surge. Fuse specifications of each model are shown as below.

Type	Fuse Type	Reference and Rating
G	Time-Lag	CONQUE MST, 6.3A, 250V
L	Time-Lag	CONQUE MST, 3.15A, 250V
H	Time-Lag	CONQUE MST, 2A, 250V

■ Input Reverse Polarity Protection

There is a MOSFET connected in series to the negative input line. If the input polarity is connected reversely, the MOSFET opens and there will be no output to protect the unit.

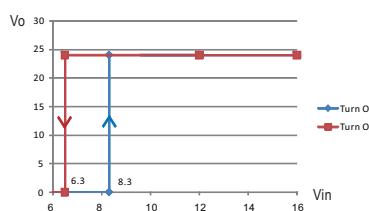
■ Input Range and Transient Ability

The series has a wide range input capability. With $\pm 40\%$ of rated input voltage, it can withstand that for 1 second.

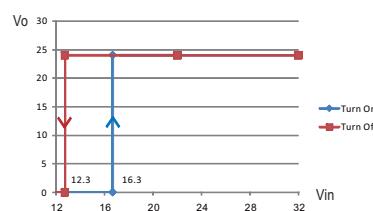
■ Input Under-Voltage Protection

If input voltage drops below V_{min} , the internal control IC shuts down and there is no output voltage. It recovers automatically when input voltage reaches above V_{min} , please refer to the curve below.

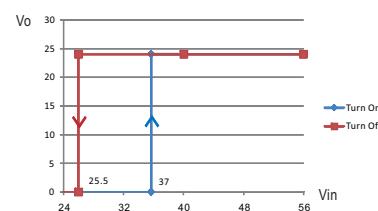
RSD-30G-24



RSD-30L-24



RSD-30H-24



■ Inrush Current

Inrush current is suppressed by a resistor during the initial start-up, and then the resistor is bypassed by a MOSFET to reduce power consumption after accomplishing the start-up.



30W Reliable Railway DC-DC Converter

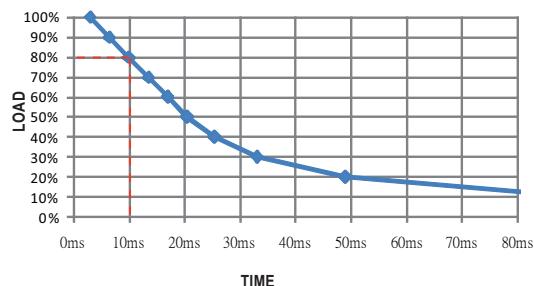
RSD-30 series

■ Hold-up Time

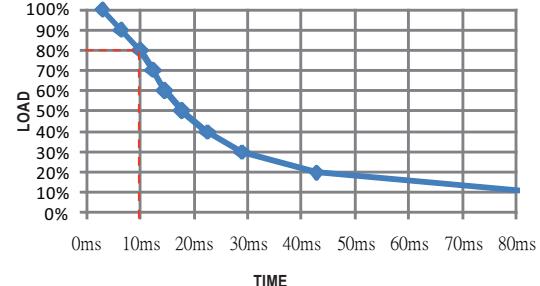
H type is in compliance with S2 level (10ms), while G and L types are in compliance with S1 level (3ms) at full load output condition.

To fulfil the requirements of S2 level (10ms), G types require de-rating their output load to 80%, please refer to the curve diagrams below.

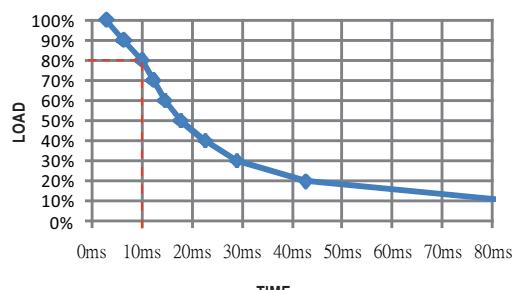
RSD-30G-3.3



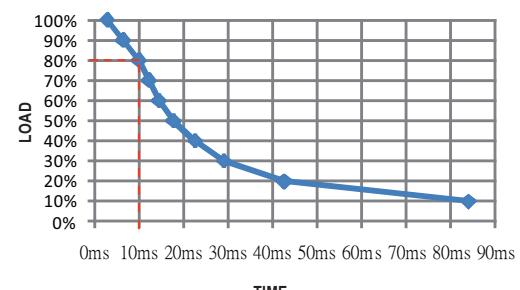
RSD-30G-5



RSD-30G-12



RSD-30G-24



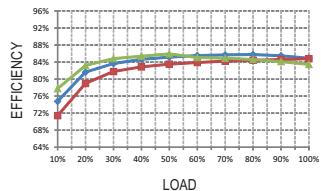
■ Output Voltage Adjustment

This function is optional, which the standard product does not have it. If you do need the function, please contact MW for details.

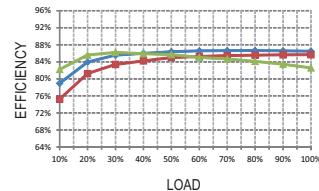
■ Efficiency vs Load & Vin Curve

The efficiency vs load & Vin curves of each model are shown as below.

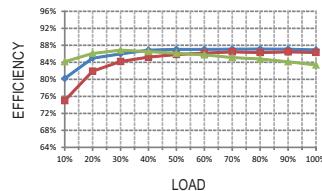
RSD-30G-3.3



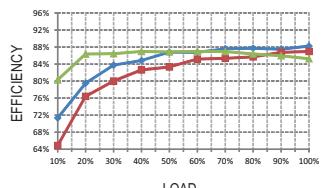
RSD-30G-5



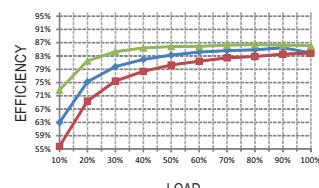
RSD-30G-12



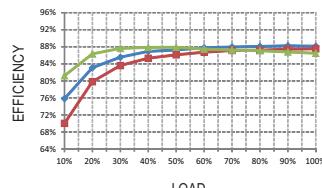
RSD-30G-24



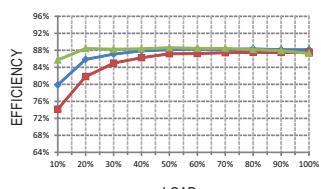
RSD-30L-3.3



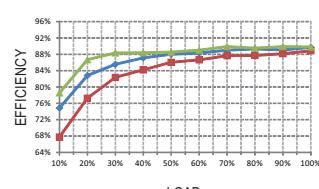
RSD-30L-5



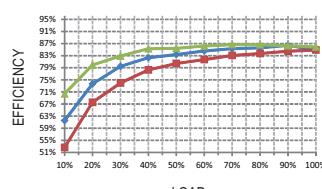
RSD-30L-12



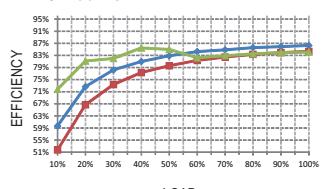
RSD-30L-24



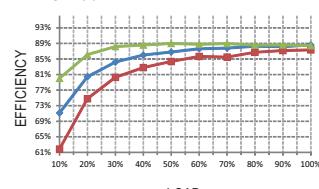
RSD-30H-3.3



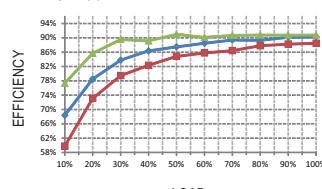
RSD-30H-5



RSD-30H-12



RSD-30H-24

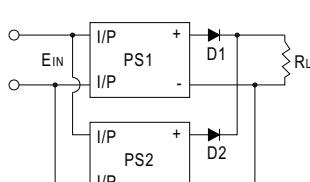


■ Parallel and Series Connection

A. Operation in Parallel

Since RSD-30 series don't have built-in parallel circuit, it can only use external circuits to achieve the redundant operation but not increase the current rating.

- Add a diode at the positive-output of each power supply (as shown as below), the current rating of the diode should be larger than the maximum output current rating and attached to a suitable heat sink. This is only for redundant use (increase the reliability of the system) and users have to check suitability of the circuit by themselves.

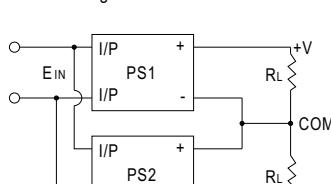


- When using S.P.S. in parallel connection, the leakage current will increase at the same time. This could pose as a shock hazard for the user. So please contact the supplier if you have this kind of application.

B. Operation in Series

RSD-30 can be operated in series. Here are the methods of doing it:

- Positive and negative terminals are connected as shown as below. According to the connection, you can get the positive and negative output voltages for your loads.

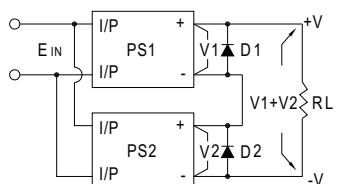




30W Reliable Railway DC-DC Converter

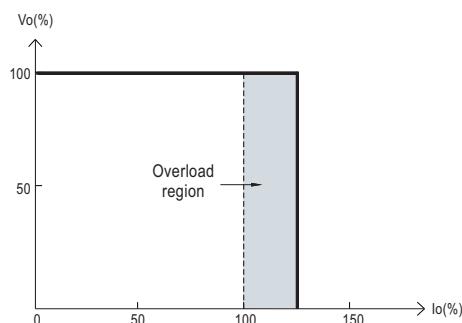
RSD-30 series

2.Increase the output voltage (current does not change). Because RSD-30 series have no reverse blocking diode in the unit, you should add an external blocking diode to prevent the damage of every unit while starting up. The voltage rating of the external diode should be larger than V_1+V_2 (as shown as below).



■ Overload Protection

If the output draw up to 105~135% of its output power rating, the converter will go into overload protection which is constant current mode. After the faulty condition is removed, it will recover automatically. Please refer to the diagram below for the detail operation characteristic. Please note that it's not suitable to operate within the overload region continuously, or it may cause to over temperature and reduce the life of the power supply unit or even damage it.



■ Over Voltage Protection

The converter shuts off to protect itself when the output voltage drawn exceeds 115~140% of its output rating. It must be repowered on to recover.

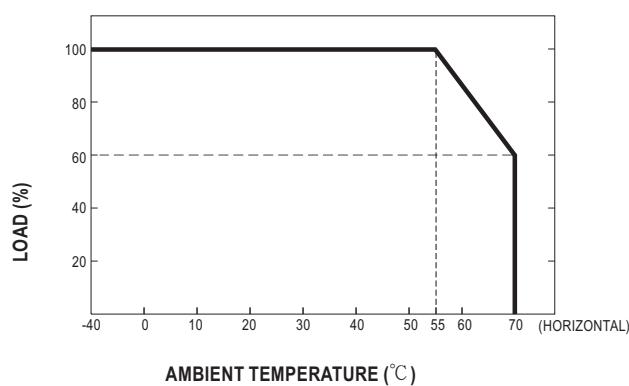
■ LED Indicator

Equipped with a built-in LED indicator, the converter provides an easy way for users to check its condition through the LED indicator.
Green : normal operation; No signal: no power or failure.

■ Derating Curve

a.Single unit operation

If the unit has no iron plate mounted on its bottom, the maximum ambient temperature for the unit will be 55°C as operating under full load condition. It requires de-rating output current when ambient temperature is between $55\text{--}70^{\circ}\text{C}$, please refer to the de-rating curve as below.

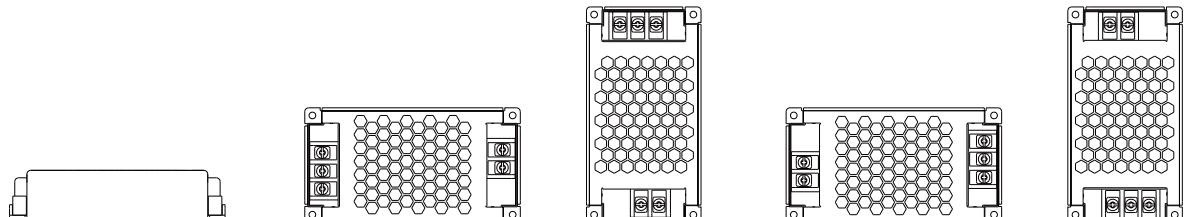




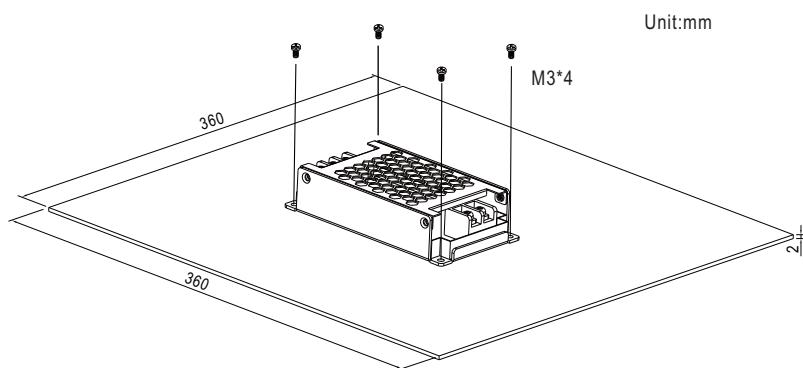
30W Reliable Railway DC-DC Converter

RSD-30 series

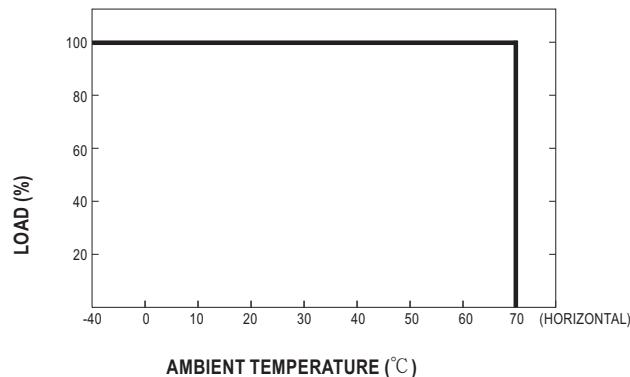
Suitable installation methods are shown as below. Since RSD-30 is a semi-potted model, its thermal performances for the following installation methods are similar and share the same derating curve.

**b. Operate with additional iron plate**

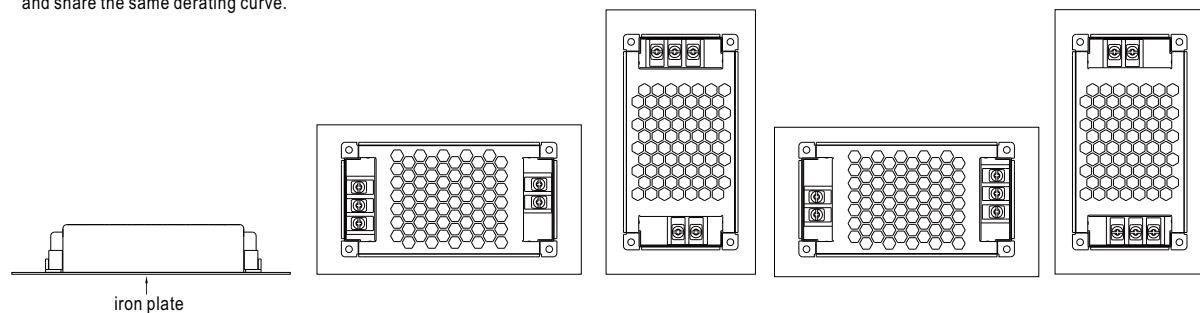
If it is necessary to fulfil the requirements of EN50155 TX level that operate the unit fully-loaded at 70°C, RSD-30 series must be installed onto an iron plate on the bottom. The size of the suggested iron plate is shown as below. In order for optimal thermal performance, the iron plate must have an even & smooth surface and RSD-30 series must be firmly mounted at the center of the iron plate.



The load vs ambient temperature curve is shown as below.



Suitable installation methods are shown as below. Since RSD-30 is a semi-potted model, its thermal performances for the following installation methods are similar and share the same derating curve.





30W Reliable Railway DC-DC Converter

RSD-30 series

■ Immunity to Environmental Conditions

Test method	Standard	Test conditions	Status
Cooling Test	EN 50155 section 12.2.3 (Column 2, Class TX) EN 60068-2-1	Temperature: -40°C Dwell Time: 2 hrs/cycle	No damage
Dry Heat Test	EN 50155 section 12.2.4 (Column 2, Class TX) EN 50155 section 12.2.4 (Column 3, Class TX & Column 4, Class TX) EN 60068-2-2	Temperature: 70°C / 85°C Duration: 6 hrs / 10min	PASS
Damp Heat Test, Cyclic	EN 50155 section 12.2.5 EN 60068-2-30	Temperature: 25°C ~ 55°C Humidity: 90% ~ 100% RH Duration: 48 hrs	PASS
Vibration Test	EN 50155 section 12.2.11 EN 61373	Temperature: 19°C Humidity: 65% Duration: 10 mins	PASS
Increased Vibration Test	EN 50155 section 12.2.11 EN 61373	Temperature: 19°C Humidity: 65% Duration: 5 hrs	PASS
Shock Test	EN 50155 section 12.2.11 EN 61373	Temperature: 21 ± 3°C Humidity: 65 ± 5% Duration: 30ms * 18	PASS
Low Temperature Storage Test	EN 50155 section 12.2.3 (Column 2, Class TX) EN 60068-2-1	Temperature: -40°C Dwell Time: 16 hrs	PASS
Salt Mist Test	EN 50155 section 12.2.10 (Class ST4)	Temperature: 35°C ± 2°C Duration: 96 hrs	PASS

■ EN45545-2 Fire Test Conditions

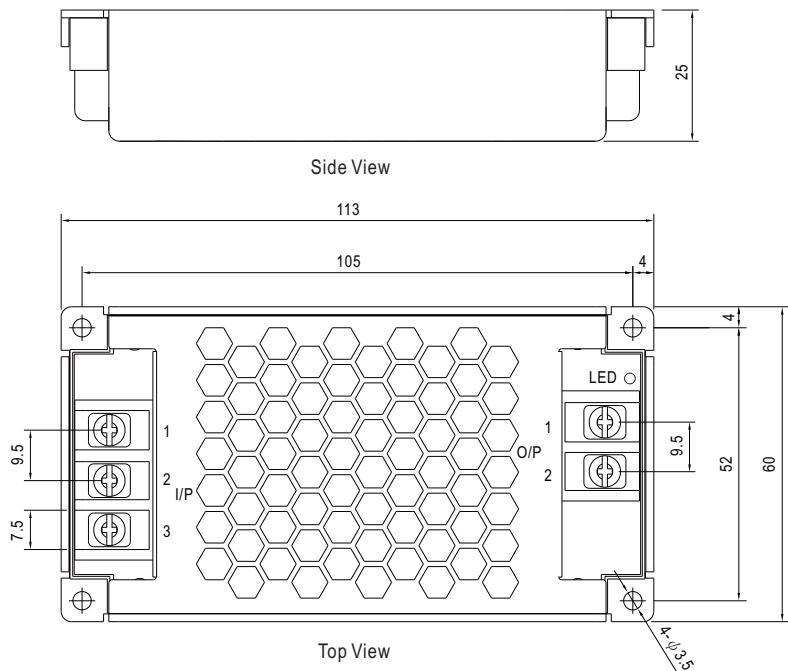
Test Items		Hazard Level			
Items		Standard	HL1	HL2	
R24	Oxygen index test	EN 45545-2:2013+A1:2015 EN ISO 4589-2:1996	PASS	PASS	PASS
R25	Glow-wire test	EN 45545-2:2013+A1:2015 EN 60695-2-11:2000	PASS	PASS	PASS
R26	Vertical flame test	EN 45545-2:2013+A1:2015 EN 60695-11:2003	PASS	PASS	PASS



30W Reliable Railway DC-DC Converter

RSD-30 series**■ Mechanical Specification**

Case No.253A Unit:mm



Input Terminal Pin No. Assignment :

Pin No.	Assignment
1	DC INPUT V+
2	DC INPUT V-
3	FG \pm

Output Terminal Pin No. Assignment :

Pin No.	Assignment
1	DC OUTPUT -V
2	DC OUTPUT +V

■ Installation ManualPlease refer to : <http://www.meanwell.com/manual.html>

0.2 Mean Well SD-350C-12



350W Single Output DC-DC Converter

SD-350 series

■ Features :

- 2:1 wide input range
- Protections: Short circuit / Overload / Over voltage / Over temperature
- 1500VAC I/O isolation
- Forced air cooling by built-in DC Fan
- 100% full load burn-in test
- 24V and 48V input voltage design refer to LVD
- 2 years warranty



EN62368-1



IEC62368-1

**SPECIFICATION**

MODEL	SD-350B				SD-350C				
OUTPUT	DC VOLTAGE	5V	12V	24V	48V	5V	12V	24V	48V
	RATED CURRENT	57A	27.5A	14.6A	7.3A	60A	27.5A	14.6A	7.3A
	CURRENT RANGE	0 ~ 57A	0 ~ 27.5A	0 ~ 14.6A	0 ~ 7.3A	0 ~ 60A	0 ~ 27.5A	0 ~ 14.6A	0 ~ 7.3A
	RATED POWER	285W	330W	350.4W	350.4W	300W	330W	350.4W	350.4W
	RISSLE & NOISE (max.) Note.2	100mVp-p	120mVp-p	150mVp-p	200mVp-p	100mVp-p	120mVp-p	150mVp-p	200mVp-p
	VOLTAGE ADJ. RANGE	4.5 ~ 5.5VDC	11 ~ 16VDC	23 ~ 30VDC	43 ~ 53VDC	4.5 ~ 5.5VDC	11 ~ 16VDC	23 ~ 30VDC	43 ~ 53VDC
	VOLTAGE TOLERANCE Note.3	±2.0%	±1.0%	±1.0%	±1.0%	±2.0%	±1.0%	±1.0%	±1.0%
	LINER REGULATION	±0.5%	±0.3%	±0.2%	±0.2%	±0.5%	±0.3%	±0.2%	±0.2%
INPUT	LOAD REGULATION	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%
	SETUP, RISE TIME	300ms, 50ms at full load							
PROTECTION	VOLTAGE RANGE	B:19 ~ 36VDC	C:36 ~ 72VDC	D:72 ~ 144VDC					
	EFFICIENCY (Typ.)	74%	80%	80%	84%	76%	81%	81%	82%
	DC CURRENT (Typ.)	14.4A/24V	16A/24V	17.6A/24V	17.6A/24V	7.6A/48V	8.8A/48V	9.0A/48V	9.0A/48V
	INRUSH CURRENT (Typ.)	C:45A/48VDC D:45A/96VDC							
ENVIRONMENT	OVERLOAD	105 ~ 135% rated output power Protection type : Shut down o/p voltage, re-power on to recover							
	OVER VOLTAGE	5.75 ~ 6.75V	16.8 ~ 20V	31.5 ~ 37.5V	53 ~ 65V	5.75 ~ 6.75V	16.8 ~ 20V	31.5 ~ 37.5V	53 ~ 65V
	OVER TEMPERATURE	Protection type : Shut down o/p voltage, re-power on to recover							
SAFETY & EMC (Note 4)	WORKING TEMP.	-20 ~ +60°C (Refer to "Derating Curve")							
	WORKING HUMIDITY	20 ~ 90% RH non-condensing							
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH							
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)							
	VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, 60min. each along X, Y, Z axes							
OTHERS	SAFETY STANDARDS	IEC62368-1 CB approved by TUV (for D type only), EAC TP TC 004 approved							
	WITHSTAND VOLTAGE	I/P-O/P:1.5KVAC I/P-FG:2KVAC O/P-FG:0.5KVAC							
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C/ 70% RH							
NOTE	EMC EMISSION	Compliance to EN55032 (CISPR32) Class B, EAC TP TC 020							
	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,6,8, light industry level, criteria A, EAC TP TC 020							
	MTBF	209.4 hrs min. MIL-HDBK-217F (25°C)							
DIMENSION	215*115*50mm (L*W*H)								
	PACKING	1.1Kg; 12pcs/14.4Kg/0.92CUFT							
1. All parameters NOT specially mentioned are measured at 24,48,96VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. The ambient temperature derating of 3.5°C/1000m with fanless models and of 5°C/1000m with fan models for operating altitude higher than 2000m(6500ft)									

File Name:SD-350-SPEC 2020-03-37



350W Single Output DC-DC Converter

SD-350 series



■ Features :

- 2:1 wide input range
- Protections: Short circuit / Overload / Over voltage / Over temperature
- 1500VAC I/O isolation
- Forced air cooling by built-in DC Fan
- 100% full load burn-in test
- 24V(B) and 48V(C) input voltage design refer to LVD
- 2 years warranty

CB (for D type only)
IEC62368-1**SPECIFICATION**

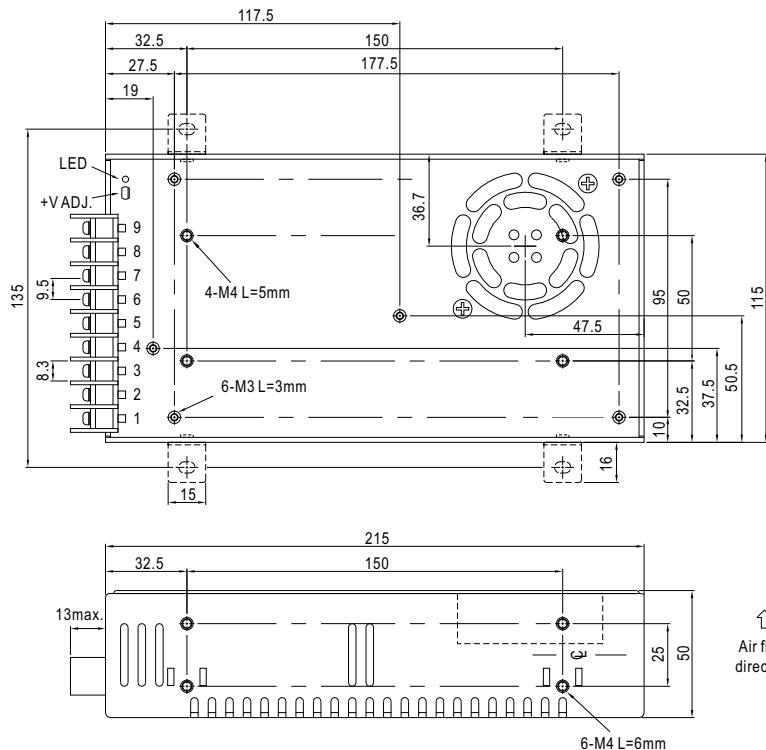
MODEL	SD-350D			
OUTPUT	DC VOLTAGE	5V	12V	24V
	RATED CURRENT	60A	29.2A	14.6A
	CURRENT RANGE	0 ~ 60A	0 ~ 29.2A	0 ~ 14.6A
	RATED POWER	300W	350.4W	350.4W
	RIPPLE & NOISE (max.) Note.2	100mVp-p	120mVp-p	150mVp-p
	VOLTAGE ADJ. RANGE	4.5 ~ 5.5VDC	11 ~ 16VDC	23 ~ 30VDC
	VOLTAGE TOLERANCE Note.3	±2.0%	±1.0%	±1.0%
	LINE REGULATION	±0.5%	±0.3%	±0.2%
	LOAD REGULATION	±1.0%	±1.0%	±1.0%
	SETUP, RISE TIME	300ms, 50ms at full load		
INPUT	VOLTAGE RANGE	B:19 ~ 36VDC	C:36 ~ 72VDC	D:72 ~ 144VDC
	EFFICIENCY (Typ.)	78%	83%	87%
	DC CURRENT (Typ.)	6A/96V	6A/96V	6A/96V
	INRUSH CURRENT (Typ.)	C:45A/48VDC	D:45A/96VDC	
PROTECTION	OVERLOAD	105 ~ 135% rated output power		
		Protection type : Shut down o/p voltage, re-power on to recover		
	OVER VOLTAGE	5.75 ~ 6.75V	16.8 ~ 20V	31.5 ~ 37.5V
		Protection type : Shut down o/p voltage, re-power on to recover		
ENVIRONMENT	OVER TEMPERATURE	Shut down o/p voltage, recovers automatically after temperature goes down		
	WORKING TEMP.	-20 ~ +60°C (Refer to "Derating Curve")		
	WORKING HUMIDITY	20 ~ 90% RH non-condensing		
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH		
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)		
SAFETY & EMC (Note 4)	VIBRATION	10 ~ 500Hz, 2G 10min./cycle, 60min. each along X, Y, Z axes		
	SAFETY STANDARDS	IEC62368-1 CB approved by TUV (for D type only), EAC TP TC 004 approved		
	WITHSTAND VOLTAGE	I/P-O/P:1.5KVAC	I/P-FG:2KVAC	O/P-FG:0.5KVAC
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C/ 70% RH		
	EMC EMISSION	Compliance to EN55022 (CISPR22) Class B, EAC TP TC 020		
OTHERS	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,6,8, light industry level, criteria A, EAC TP TC 020		
	MTBF	209.4K hrs min. MIL-HDBK-217F (25°C)		
	DIMENSION	215*115*50mm (L*W*H)		
NOTE	PACKING	1.1Kg; 12pcs/14.4Kg/0.92CUFT		
	1. All parameters NOT specially mentioned are measured at 24,48,96VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. The ambient temperature derating of 3.5°C/1000m with fanless models and of 5°C/1000m with fan models for operating altitude higher than 2000m(6500ft).			



350W Single Output DC-DC Converter

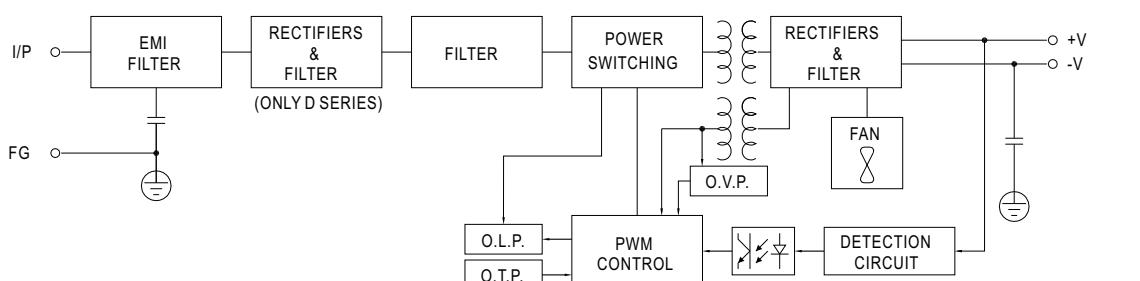
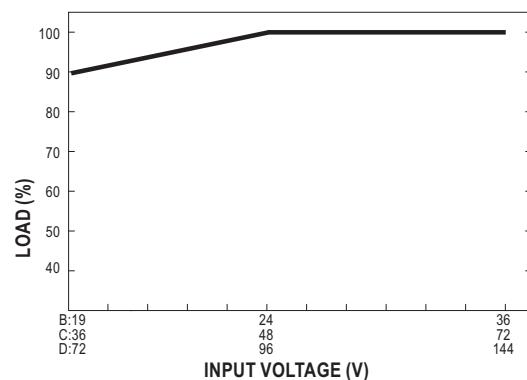
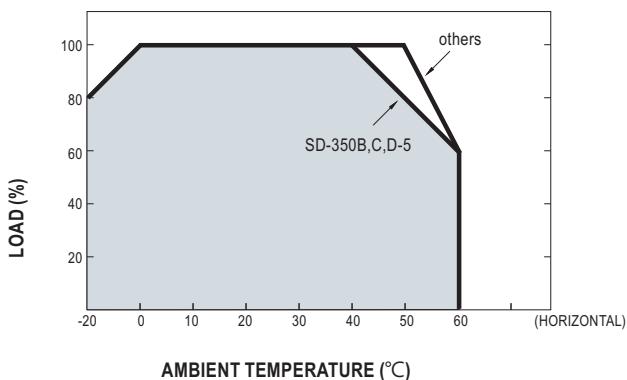
SD-350 series**Mechanical Specification**

Case No. 912B Unit:mm



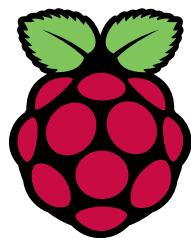
Terminal Pin No. Assignment :

Pin No.	Assignment	Pin No.	Assignment
1	DC INPUT V+	4,5,6	DC OUTPUT V-
2	DC INPUT V-	7,8,9	DC OUTPUT V+
3	FG \pm		

Block Diagram**Derating Curve****Static Characteristics**

0.3 Raspberry Pi 4 Moddel B

DATASHEET



Raspberry Pi 4 Model B

Release 1

June 2019

Copyright 2019 Raspberry Pi (Trading) Ltd. All rights reserved.



Raspberry Pi 4 Model B Datasheet
Copyright Raspberry Pi (Trading) Ltd. 2019

Table 1: Release History

Release	Date	Description
1	21/06/2019	First release

The latest release of this document can be found at <https://www.raspberrypi.org>



Contents

1	Introduction	5
2	Features	6
2.1	Hardware	6
2.2	Interfaces	6
2.3	Software	7
3	Mechanical Specification	7
4	Electrical Specification	7
4.1	Power Requirements	9
5	Peripherals	9
5.1	GPIO Interface	9
5.1.1	GPIO Pin Assignments	9
5.1.2	GPIO Alternate Functions	10
5.1.3	Display Parallel Interface (DPI)	11
5.1.4	SD/SDIO Interface	11
5.2	Camera and Display Interfaces	11
5.3	USB	11
5.4	HDMI	11
5.5	Audio and Composite (TV Out)	11
5.6	Temperature Range and Thermals	11
6	Availability	12
7	Support	12



List of Figures

1	Mechanical Dimensions	7
2	Digital IO Characteristics	8
3	GPIO Connector Pinout	9



List of Tables

1	Release History	1
2	Absolute Maximum Ratings	8
3	DC Characteristics	8
4	Digital I/O Pin AC Characteristics	8
5	Raspberry Pi 4 GPIO Alternate Functions	10



1 Introduction

The Raspberry Pi 4 Model B (Pi4B) is the first of a new generation of Raspberry Pi computers supporting more RAM and with significantly enhanced CPU, GPU and I/O performance; all within a similar form factor, power envelope and cost as the previous generation Raspberry Pi 3B+.

The Pi4B is available with either 1, 2 and 4 Gigabytes of LPDDR4 SDRAM.



2 Features

2.1 Hardware

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 1, 2 and 4 Gigabyte LPDDR4 RAM options
- H.265 (HEVC) hardware decode (up to 4Kp60)
- H.264 hardware decode (up to 1080p60)
- VideoCore VI 3D Graphics
- Supports dual HDMI display output up to 4Kp60

2.2 Interfaces

- 802.11 b/g/n/ac Wireless LAN
- Bluetooth 5.0 with BLE
- 1x SD Card
- 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution
- 2x USB2 ports
- 2x USB3 ports
- 1x Gigabit Ethernet port (supports PoE with add-on PoE HAT)
- 1x Raspberry Pi camera port (2-lane MIPI CSI)
- 1x Raspberry Pi display port (2-lane MIPI DSI)
- 28x user GPIO supporting various interface options:
 - Up to 6x UART
 - Up to 6x I2C
 - Up to 5x SPI
 - 1x SDIO interface
 - 1x DPI (Parallel RGB Display)
 - 1x PCM
 - Up to 2x PWM channels
 - Up to 3x GPCLK outputs



2.3 Software

- ARMv8 Instruction Set
- Mature Linux software stack
- Actively developed and maintained
 - Recent Linux kernel support
 - Many drivers upstreamed
 - Stable and well supported userland
 - Availability of GPU functions using standard APIs

3 Mechanical Specification

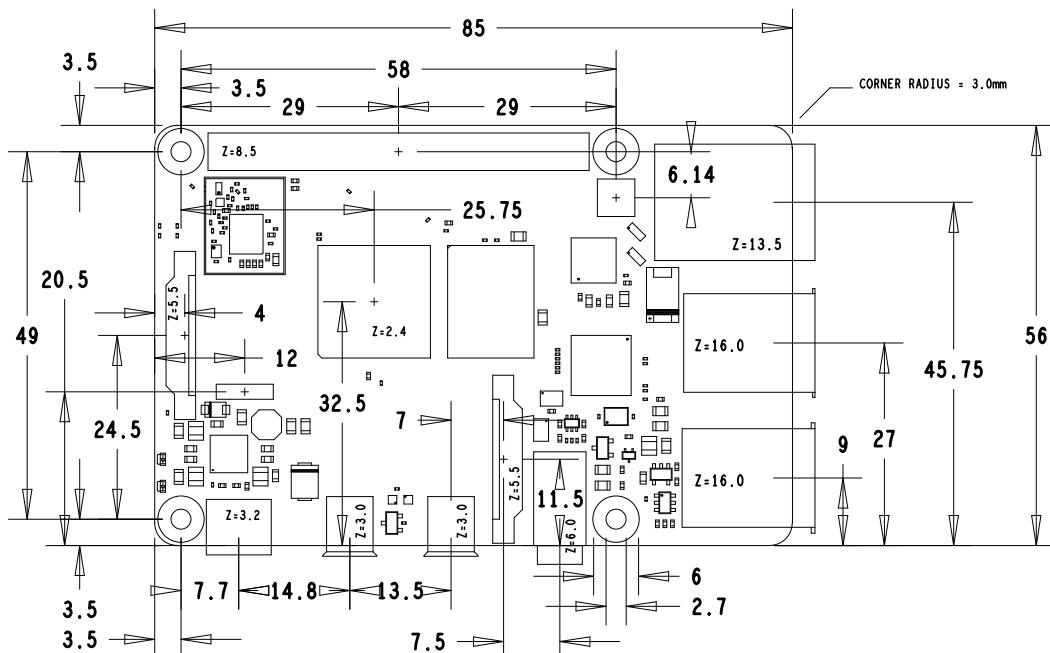


Figure 1: Mechanical Dimensions

4 Electrical Specification

Caution! Stresses above those listed in Table 2 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.


 Raspberry Pi 4 Model B Datasheet
 Copyright Raspberry Pi (Trading) Ltd. 2019

Symbol	Parameter	Minimum	Maximum	Unit
VIN	5V Input Voltage	-0.5	6.0	V

Table 2: Absolute Maximum Ratings

Please note that VDD_IO is the GPIO bank voltage which is tied to the on-board 3.3V supply rail.

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Input low voltage ^a	$VDD_IO = 3.3V$	-	-	TBD	V
V_{IH}	Input high voltage ^a	$VDD_IO = 3.3V$	TBD	-	-	V
I_{IL}	Input leakage current	$TA = +85^{\circ}C$	-	-	TBD	μA
C_{IN}	Input capacitance	-	-	TBD	-	pF
V_{OL}	Output low voltage ^b	$VDD_IO = 3.3V, IOL = -2mA$	-	-	TBD	V
V_{OH}	Output high voltage ^b	$VDD_IO = 3.3V, IOH = 2mA$	TBD	-	-	V
I_{OL}	Output low current ^c	$VDD_IO = 3.3V, VO = 0.4V$	TBD	-	-	mA
I_{OH}	Output high current ^c	$VDD_IO = 3.3V, VO = 2.3V$	TBD	-	-	mA
R_{PU}	Pullup resistor	-	TBD	-	TBD	k Ω
R_{PD}	Pulldown resistor	-	TBD	-	TBD	k Ω

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Table 3: DC Characteristics

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t_{rise}	10-90% rise time ^a	-	TBD	-	ns
Digital outputs	t_{fall}	90-10% fall time ^a	-	TBD	-	ns

^a Default drive strength, $CL = 5pF, VDD_IO = 3.3V$

Table 4: Digital I/O Pin AC Characteristics



Figure 2: Digital IO Characteristics



4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.

5 Peripherals

5.1 GPIO Interface

The Pi4B makes 28 BCM2711 GPIOs available via a standard Raspberry Pi 40-pin header. This header is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

5.1.1 GPIO Pin Assignments

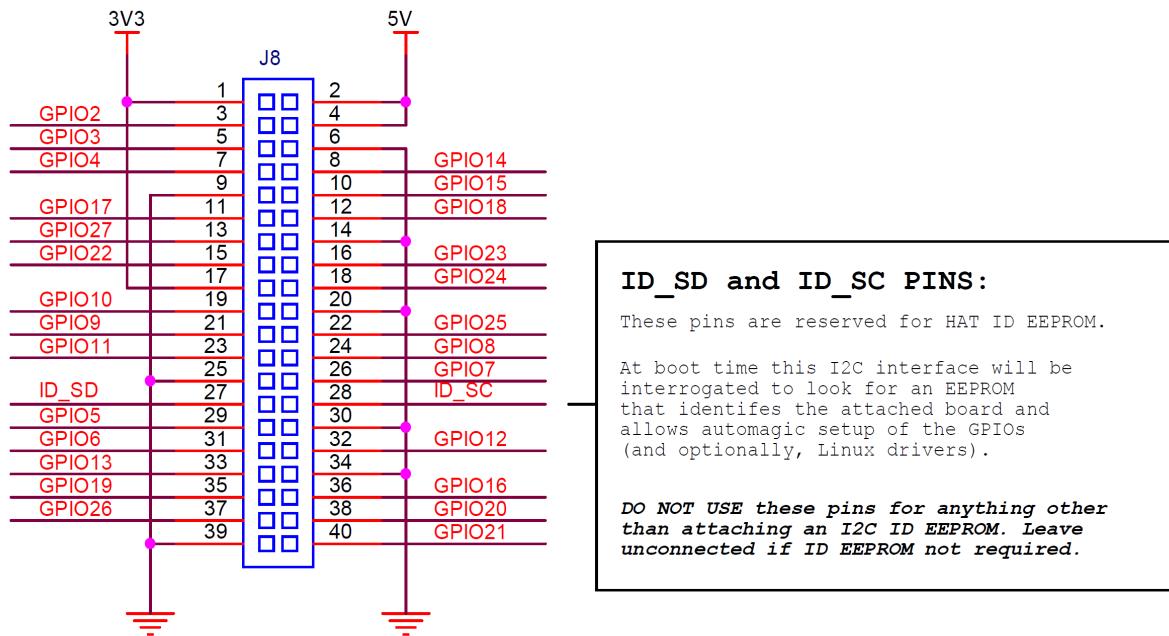


Figure 3: GPIO Connector Pinout

As well as being able to be used as straightforward software controlled input and output (with programmable pulls), GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI.

In addition to the standard peripheral options found on legacy Pis, extra I2C, UART and SPI peripherals have been added to the BCM2711 chip and are available as further mux options on the Pi4. This gives users much more flexibility when attaching add-on hardware as compared to older models.



5.1.2 GPIO Alternate Functions

GPIO	Pull	Default					
		ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SDA6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCLK0	SA1	DPI_D0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCLK1	SA0	DPI_D1	SPI4_MISO	RXD3	SCL3
6	High	GPCLK2	SOE_N	DPI_D2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPI_D3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPI_D4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPI_D5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPI_D6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPI_D7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPI_D8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPI_D9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPI_D10	SPI5_MOSI	CTS5	TXD1
15	Low	RXD0	SD7	DPI_D11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPI_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPI_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI_D14	SPI6_CE0_N	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPI_D15	SPI6_MISO	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPI_D16	SPI6_MOSI	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPI_D17	SPI6_SCLK	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPI_D18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CMD	SD15	DPI_D19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_DAT0	SD16	DPI_D20	SD1_DAT0	ARM_TDO	SPI3_CE1_N
25	Low	SD0_DAT1	SD17	DPI_D21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT2	TE0	DPI_D22	SD1_DAT2	ARM_TDI	SPI5_CE1_N
27	Low	SD0_DAT3	TE1	DPI_D23	SD1_DAT3	ARM_TMS	SPI6_CE1_N

Table 5: Raspberry Pi 4 GPIO Alternate Functions

Table 5 details the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the BCM2711 Peripherals Specification document which can be downloaded from the hardware documentation section of the website.



5.1.3 Display Parallel Interface (DPI)

A standard parallel RGB (DPI) interface is available the GPIOs. This up-to-24-bit parallel interface can support a secondary display.

5.1.4 SD/SDIO Interface

The Pi4B has a dedicated SD card socket which supports 1.8V, DDR50 mode (at a peak bandwidth of 50 Megabytes / sec). In addition, a legacy SDIO interface is available on the GPIO pins.

5.2 Camera and Display Interfaces

The Pi4B has 1x Raspberry Pi 2-lane MIPI CSI Camera and 1x Raspberry Pi 2-lane MIPI DSI Display connector. These connectors are backwards compatible with legacy Raspberry Pi boards, and support all of the available Raspberry Pi camera and display peripherals.

5.3 USB

The Pi4B has 2x USB2 and 2x USB3 type-A sockets. Downstream USB current is limited to approximately 1.1A in aggregate over the four sockets.

5.4 HDMI

The Pi4B has 2x micro-HDMI ports, both of which support CEC and HDMI 2.0 with resolutions up to 4Kp60.

5.5 Audio and Composite (TV Out)

The Pi4B supports near-CD-quality analogue audio output and composite TV-output via a 4-ring TRS 'A/V' jack.

The analog audio output can drive 32 Ohm headphones directly.

5.6 Temperature Range and Thermals

The recommended ambient operating temperature range is 0 to 50 degrees Celcius.

To reduce thermal output when idling or under light load, the Pi4B reduces the CPU clock speed and voltage. During heavier load the speed and voltage (and hence thermal output) are increased. The internal governor will throttle back both the CPU speed and voltage to make sure the CPU temperature never exceeds 85 degrees C.

The Pi4B will operate perfectly well without any extra cooling and is designed for sprint performance - expecting a light use case on average and ramping up the CPU speed when needed (e.g. when loading a webpage). If a user wishes to load the system continually or operate it at a high temperature at full performance, further cooling may be needed.



Raspberry Pi 4 Model B Datasheet
Copyright Raspberry Pi (Trading) Ltd. 2019

6 Availability

Raspberry Pi guarantee availability Pi4B until at least January 2026.

7 Support

For support please see the hardware documentation section of the Raspberry Pi website and post questions to the Raspberry Pi forum.

Abbildungsverzeichnis

V.1	Raspberry Pi - Steuereinheit des HCIS	9
V.2	Grundaufbau des Human-Computer Interaction Systems	9
V.3	Anschlussplan Eingänge	11
V.4	Anschlussplan Relais	12
V.5	Paneel Maße	13
V.6	Befestigung des Displays	14
V.7	Aufbau der Graphischen Benutzeroberfläche	14
V.8	GUI Komponente - Navigation Menü	15
V.9	GUI Komponente - Balken Anzeige	16
V.10	GUI Komponente - Modus Anzeige	16
V.11	GUI Komponente - Graph	17
V.12	GUI Fenster - Login Menü	18
V.13	GUI Fenster - Fahrdaten	18
V.14	GUI Fenster - Akkudaten	19
V.15	GUI Fenster - Fahrdaten Diagnose	19
V.16	GUI Fenster - Fehler Liste	20
V.17	Slots und Signals Konzept	21
V.18	Verbindung Frontend zu Backend	22
V.19	Anschlussplan CAN-Modul	23

Tabellenverzeichnis

V.1	Berechnung der Leistung des 12V-Systems	10
V.2	Datenbankstruktur der Benutzer Tabelle	25
V.3	Datenbankstruktur der Datenpaket 1 Tabelle	25
V.4	Datenbankstruktur der Datenpaket 2 Tabelle	26
V.5	Datenbankstruktur der Datenpaket 3 Tabelle	26
V.6	Datenbankstruktur der Fehler-Datenpaket Tabelle	27

Code Listings

V.1	Code zum Starten eines Threads	12
V.2	Konfigurieren des CAN Adapters	24
V.3	Konfiguration der Datenbankschnittstelle	27
V.4	Konfiguration der Datenbankschnittstelle	28