

Das Pflichtenheft beinhaltet **möglichst konkret** die Umsetzung der Kundenanforderungen (Lastenheft); also nur die Umsetzung der Anforderungen, die tatsächlich realisiert werden.

Dazu sollten auch Visualisierungsmöglichkeiten (Screenshots, Zeichnungen, etc.) eingesetzt werden.

Das fertige Pflichtenheft dient als Vertragsgrundlage (s. u.)!

<b>Pflichtenheft</b> (DIN VDI/VDE 3694 / ISO 9002)	
<b>Projekt:</b>	Werkstatttorüberwachung beim Autohaus Nettmann
<b>Autor:</b>	Auftragnehmer (Schüler-Gruppe)
<b>Version:</b>	1.0
<b>Letzte Änderung:</b>	xx.xx.20xx
<b>Dateiablage:</b>	
<b>Ziel des Systems:</b>	Realisierung einer IT-gestützten Überwachung der vier Werkstatttore des AH Nettmann. Aufgrund versicherungsrechtlicher Vorgaben muss sichergestellt sein, dass die Tore nach Arbeitsschluss geschlossen sind. Eine Überwachung mit mobilen Endgeräten und Speicherung der Zustände soll deshalb realisiert werden.
<b>Anforderungsliste:</b>	
<i>Produktfunktionen:</i>	Die Umsetzung der genauen funktionalen Anforderungen hinsichtlich Benutzerinterfaces etc. ist den beigefügten oder noch zu erstellenden Unterlagen (Mockups, Struktogramme, UML-Diagrammen, etc.) zu entnehmen.
1.1	Der Zustand der Werkstatttore (Offen/Geschlossen) wird über einen Mikroschalter (Schließer) erfasst.
1.2	Der Zustand des Mikroschalters (Stromkreis geschlossen □ Tor geschlossen, Stromkreis unterbrochen □ Tor offen) wird von den GPIO-Eingängen eines Raspberry PI V3 erfasst.
1.3	Der Zustand der Werkstatttore soll außerdem direkt am Raspberry über vier LEDs an GPIO-Ausgängen angezeigt werden.
1.4	Die Zustandsänderungen sollen innerhalb einer Sekunde angezeigt und erfasst werden.
1.5	Jede Zustandsänderung der Werkstatttore soll in einer Tabelle einer Datenbank abgespeichert und mit einem sekundengenauen Zeitstempel versehen werden.
1.6	Außerdem soll (wie in 1.5) jede Zustandsänderung in einer Cloud-basierten NoSQL-Datenbank gespeichert werden.
1.7	Die aktuellen Zustände der Werkstatttore soll über mobile Geräte (Laptop, Handy, ...) innerhalb und außerhalb der Firma angezeigt werden können.

1.8	Security-Maßnahmen sind mittelfristig unbedingt zu realisieren, sollen aber in einem ersten Schritt erst einmal nur beschrieben und dokumentiert werden.
<i>Produktdaten:</i>	Die Bezeichnung der Variablen und die GPIO-Belegung des Raspberry PI sind dem Anhang zu entnehmen. Als Datentyp für die Zustände der Werkstatttore ist Boolean, für den Zeitstempel ein sekundengenaue Timestamp-Datentyp zu wählen.
<i>Leistungsanforderungen/ Rahmenbedingungen:</i>	
2.1	Für die Realisierung des Teilsystems „Werkstatttorüberwachung“ ist ein Raspberry PI V3 einzusetzen.
2.2	Die Anwendung des Teilsystems „Werkstatttorüberwachung“ ist in Python zu realisieren. Das Programm zur Werkstatttorüberwachung soll nach einem Stromausfall selbständig wieder starten.
2.3	Speicherung der Zustandsdaten <b>intern</b> in einer MySQL-Server-Datenbank auf Windows10.
2.4	Speicherung der Zustandsdaten <b>extern</b> in der cloudbasierten NoSQL-Datenbank MongoDB (DBaaS).
2.5	Der Transport der Zustände der Werkstatttore erfolgt über die Kommunikationsprotokolle OPC UA oder MQTT.
2.6	Als Middleware-Software wird der OPC Router der Firma Inray verwendet.
<b>Termin der Abnahme</b>	Fertigstellung (Realisierung der Werkstatttorüberwachung) bis: _____

Die unterschreibenden Personen versichern, dass die in diesem Dokument aufgestellten Anforderungen das zu entwickelnde System vollständig beschreiben. Es sind zum Zeitpunkt der Unterschriftsleistung keine weiteren Anforderungen bekannt, die nicht in diesem Dokument beschrieben wurden.

Es gelten keinerlei Anforderungen, die in weiteren Dokumenten beschrieben werden, außer diese detaillierten im Pflichtenheft beschriebenen Anforderungen. Zusätzliche Anforderungen oder Änderungen an den bestehenden Anforderungen bedürfen der Schriftform.

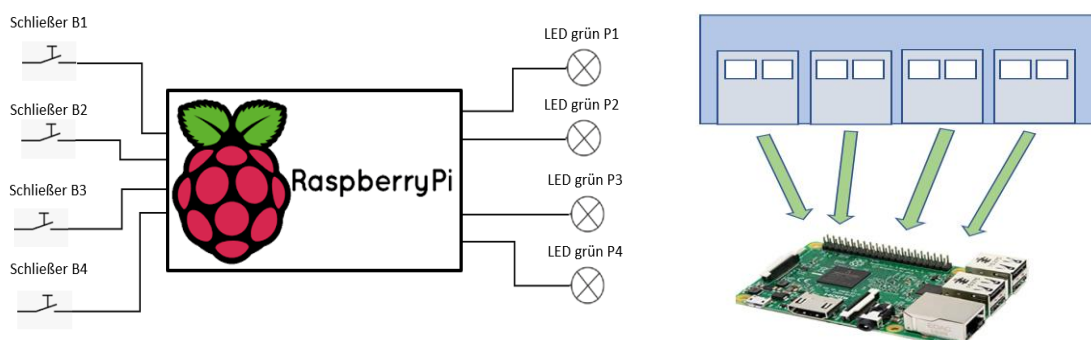
Projekt-Auftraggeber \_\_\_\_\_

Ort, Datum, Unterschrift

Projektleiter \_\_\_\_\_

Ort, Datum, Unterschrift

## Anhang 1: Überblick Werkstatttorüberwachung



## Anhang 2: Zuordnungstabelle für Rasp PI

Zuordnungstabelle				
	Symbol	Operand (Python-Variable)	Raspi BCM/ Pin	Kommentar
Eingang (Schließer/Taster)	B1	I1 (S1)	GPIO 5	Taster (Schließer1)
	B2	I2 (S2)	GPIO 6	Taster (Schließer2)
	B3	I3 (S3)	GPIO 13	Taster (Schließer3)
	B4	I4 (S4)	GPIO 19	Taster (Schließer4)
Ausgang (LED)	P1	Q1 (LED grün) (a[0]) Array	GPIO 16	Status LED
	P2	Q2 (LED grün) (a[1]) Array	GPIO 12	Status LED
	P3	Q3 (LED grün) (a[2]) Array	GPIO 7	Status LED
	P4	Q4 (LED grün) (a[3]) Array	GPIO 8	Status LED

## Anhang 3: Zuordnungstabelle für Rasp PI (Board Fa. Bernhardt, Schaltplan Anhang5)

Zuordnungstabelle				
	Symbol	Operand (Python-Variable)	Raspi BCM/ Pin	Kommentar
Eingang (Schließer/Taster)	IX02	I1 (S1)	GPIO ?	Taster (Schließer1)
	IX03	I2 (S2)	GPIO ?	Taster (Schließer2)
	IX04	I3 (S3)	GPIO ?	Taster (Schließer3)
	IX05	I4 (S4)	GPIO ?	Taster (Schließer4)
Ausgang (LED)	QX02	Q1 (LED grün) (a[0]) Array	GPIO ?	Status LED
	QX03	Q2 (LED grün) (a[1]) Array	GPIO ?	Status LED
	QX04	Q3 (LED grün) (a[2]) Array	GPIO ?	Status LED
	QX05	Q4 (LED grün) (a[3]) Array	GPIO ?	Status LED

## Anhang 4: Funktionstabelle für die Werkstatttorüberwachung

I1	I2	I3	I4	Q1 (grün)	Q2 (grün)	Q3 (grün)	Q4 (grün)
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1

1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1

## Anhang 5:

