**Projekt**

**Sensorbox**

**Pflichtenheft**

b2 electronics GmbH

Riedstraße 1

6833 Klaus

Austria

**Auftraggeber**: b2 electronics GmbH

**Projektleiter**: Linus Marte

**Projektteam**: Alexander Schatzmann, Phillip Mayer, Mert Yilmaz

**Sonstige Beteiligte:** Lampert Philipp

**Projektnummer**: 0001

Dieses Pflichtenheft ist Eigentum der b2 electronics GmbH  
und darf ohne Genehmigung der Geschäfts­führung   
weder vervielfältigt oder Dritten zur Einsichtnahme  
überlassen werden. Es ist urheberrechtlich geschützt.

**Revisionsstand Pflichtenheft**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Datum** | **Name** | **Änderungen** |
| A | 4.08.2024 | LM | Erstellung der 1. Version des Pflichtenhefts |
| B | 16.09.2024 | LM | Überarbeitung |
|  |  |  |  |
|  |  |  |  |

**Inhalt**

[1. Projektübersicht 3](#_Toc177904479)

[2. Projektbeschreibung 4](#_Toc177904480)

[3. Technik 7](#_Toc177904481)

[4. Projektkosten 14](#_Toc177904482)

[5. Arbeitsplan / Meilensteine 14](#_Toc177904483)

[6. Design-Verifizierung 15](#_Toc177904484)

[7. Prototypen / Serienfreigabe 16](#_Toc177904485)

[8. Änderungsvermerke Projekt / Serienstand 17](#_Toc177904486)

[9. Kennzahlen 18](#_Toc177904487)

[10. Gesprächsprotokolle 18](#_Toc177904488)

# Projektübersicht

## Projektbudget

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressourcen-/Kostenart** | Kosten geplant (in €) | Kosten Ist (in €) | Zeit geplant (h) | Zeit Ist (h) |
| Personal | - | - | - | - |
| Material | 200-300€ | - | - | - |
| Externe Kosten | - | - | - | - |
| Investitionen | - | - | - | - |
| Sonstige Kosten | 100€ | - | - | - |
| **Gesamt** | 300-400€ | - | - | - |

Achtung: Die Preise sind nur geschätzt und können noch Variieren.

## Terminplan - Hardware

|  |  |  |
| --- | --- | --- |
| **Bezeichnung** | **Termin geplant (Monat/Jahr)** | **Termin ist (Monat Jahr)** |
| Vorprojekt/Projekt-Start | August-2024 | August-2024 |
| Funktionsmuster | Oktober-2024 |  |
| Prototypen | November-2024 |  |
| Verbesserungen | November-2024 |  |
| Testen und Gehäuse | Dezember-2024 |  |
| Projektende | 21.3.2025 |  |

## Terminplan - Software

|  |  |  |
| --- | --- | --- |
| **Bezeichnung** | **Termin geplant (Monat/Jahr)** | **Termin ist (Monat Jahr)** |
| Vorprojekt/Projekt-Start | 8.2024 |  |
| Bibliotheken/Grundsatz | 10.2024 |  |
| Test mit Prototyp | 11.2024 |  |
| Überarbeitung Ende | 12.2024 |  |
| Doku und Fertigstellung | 2.2025 |  |
| Projektende | 21.3.2025 |  |

## Freigabe Pflichtenheft

|  |  |
| --- | --- |
| **Freigabe Geschäftsleitung (F4)** | |
| siehe Freigabeblatt |  |

# Projektbeschreibung

## Ausgangssituation

Präzise Messdaten zu erfassen ist ein wichtiger Schritt zur Identifizierung von Verbesserungspotentialen oder Schwachstellen in Systemen. Besonders bei der Entwicklung, elektronischer Geräte und Komponenten ist die Erfassung von Messdaten wichtig, um das System den Anforderungen entsprechend zu entwerfen.

Oft sind Messwerte von Interesse, bei denen stationäre Messsysteme nicht verwendet werden können, z.B. an schwer zugänglichen Stellen, bei Hochspannungsanwendungen oder über sehr lange Zeiträume. Hier sind spezialisierte, miniaturisierte Datenlogger gefragt. Dabei sollte der nötige Einrichtungsaufwand minimal sein, um schnell an die gewünschten Messwerte zu gelangen.

## Zielsetzung

Ziel unserer Diplomarbeit ist es, eine benutzerfreundliche, mobile Sensorbox zu entwickeln, die präzise Messdaten auch unter extremen Bedingungen erfassen kann. Die Sensorbox soll selbstversorgt, leicht zu bedienen und mittels Mesh-Netzwerk verbindbar sein. Sie muss in einem Temperaturbereich von -20°C bis 60°C betrieben werden und mindestens 100 Stunden netzgebunden (max. 12V DC) sowie 50 Stunden netzunabhängig (Batterie / Akku) Messdaten aufzeichnen können. Die Sensorbox soll Stand-alone und im Mesh-Netzwerk mit mindestens 5 Einheiten arbeiten, kompatibel mit analogen und digitalen externen Sensoren sein und angeschlossene Sensoren automatisch erkennen und die Daten beispielsweise auf eine SD-Karte speichern können. Die Bedienelemente sollen auf 1-2 Taster und 1-2 LEDs beschränkt sein.

### **Umsetzung:**

Die Umsetzung der **Main Sensorbox** erfordert präzise Planung, um die Kommunikation und Datenerfassung effizient zu gestalten. Hier eine detaillierte Beschreibung der Umsetzungsschritte:

#### **Zentrale Steuereinheit mit ESP32**

* Die **Main Sensorbox** wird von einem **ESP32-Modul** gesteuert, welches die Aufgabe hat, Daten von mehreren peripheren ESP32-Modulen zu empfangen. Jedes periphere Modul ist mit spezifischen Sensoren (z. B. Temperatur, Feuchtigkeit, Bewegung) ausgestattet und sendet die Sensordaten drahtlos zur Main Sensorbox.
* Die Kommunikation zwischen den Modulen erfolgt vollständig über WiFi. Die Main Sensorbox fungiert als Access Point, zu dem sich die peripheren Sensorboxen direkt verbinden. Diese direkte WiFi-Verbindung ermöglicht es, dass die Sensordaten ohne zusätzliche Protokolle übertragen werden. Die Main Sensorbox nimmt die Rolle eines zentralen Knotenpunkts ein und ermöglicht es, alle verbundenen Sensorboxen in einem lokalen Netzwerk zu integrieren. Durch die Verwendung des Access Point-Modus wird eine stabile, unabhängige Netzwerkarchitektur geschaffen, die keine externe Infrastruktur wie Router benötigt.

#### **Stromversorgung und Schutz**

* Die Main Sensorbox wird über ein Netzteil oder einen Akku versorgt. Da das ESP32-Modul mit 3.3V arbeitet, muss ein **Spannungsregler** integriert werden, falls die Eingangsspannung höher ist (z. B. 5V oder 12V). Zusätzlich wird eine **Schutzdiode** zur Verhinderung von Überspannungen in den Schaltplan aufgenommen.

#### **Modularität**

* Der Schaltplan wird modular entworfen, sodass jedes periphere ESP32-Modul über definierte Schnittstellen mit der Main Sensorbox verbunden ist. Diese Modularität ermöglicht es, Sensoren einfach hinzuzufügen oder auszutauschen, ohne den gesamten Aufbau zu verändern.

#### **Datenaufzeichnung auf eine SD-Karte**

* Eine **MicroSD-Karten-Schnittstelle** wird in die Main Sensorbox integriert, um Sensordaten lokal zu speichern.
* Die Daten werden in regelmäßigen Abständen oder bei bestimmten Ereignissen auf die SD-Karte geschrieben. Dies ermöglicht eine **Offline-Datenspeicherung** und stellt sicher, dass auch bei Verbindungsproblemen alle Daten gespeichert werden.
* Die SD-Kartenanbindung erfolgt über die **SPI-Schnittstelle** des ESP32, und der Code wird entsprechend angepasst, um die Daten effizient in **CSV**- oder **TXT**-Formaten abzulegen, die später auf einem Computer ausgewertet werden können.

### **Schaltplan-Entwurf**

#### **1. Software zur Schaltplanerstellung**

* Das **Eagle – Tool** wird verwendet, um den Schaltplan präzise zu erstellen. Dieses Programm bieten umfangreiche Bibliotheken für Bauteile wie ESP32-Module, Sensoren und Displays.

#### **2. Pin-Zuordnung und Verdrahtung**

* Die Pins des ESP32 werden im Schaltplan klar den Komponenten (Sensoren, …) zugeordnet. Hierbei wird besonders auf die korrekte Stromversorgung und die Kommunikationsprotokolle geachtet.

#### **3. Testpunkte und Erweiterbarkeit**

* Im Schaltplan werden **Testpunkte** für Spannung und Signalverbindungen eingebaut, um während des Entwicklungsprozesses und später bei der Wartung einfache Messungen und Fehlerdiagnosen zu ermöglichen.

### **Sensorik:**

Dies sind die Sensoren, welche wir für unser Projekt verwenden wollen.

* **Temperatursensor:**
* Misst präzise Temperaturen in einem bestimmten Bereich.
* **Luftdruck:**
* Erfasst den atmosphärischen Druck, oft für Wetterstationen oder Höhenmessungen verwendet.
* **Feuchtigkeit**
* Misst die relative Luftfeuchtigkeit und wird in HVAC-Systemen oder zur Überwachung der Luftqualität eingesetzt.
* **Ozonmessung:**
* Bestimmt die Konzentration von Ozon in der Luft und wird zur Umweltüberwachung verwendet.

### **Auflösung:**

* Ein Beispiel dafür wäre ADC (Analog-Digital-Converter)
* Bei ADC, wird ein Analoger Wert in einen Digitalen Wert umgewandelt
* Zum Beispiel, man schickt eine Nachricht von 10Bit, mit einer Versorgungspannung von 3V
* Dort entsteht ein Fehler von: 3V/2^10
* Dieser Fehler kann die Messwerte beeinflussen, deshalb ist es ratsam, diesen so klein wie möglich zu halten

## Merkmale

* Mesh-Netzwerk – Protokoll erfolgt vollständig über WLAN
* Adaptiv: Man kann mehrere Sensoren hinzufügen
* Möglichst hohe Akkulaufzeit
* Datenspeicherung auf eine SD-Karte

# Technik

## Funktionsbeschreibung

Die Sensorbox nutzt einen ESP32 als Netzwerkhub, der als zentraler Knotenpunkt dient. Alle Sensorboxen im Netzwerk verbinden sich mit dem ESP32 und senden die aufgezeichneten Daten dorthin. Der Netzwerkhub übernimmt die Verarbeitung der bekommenen Daten und sorgt für eine effiziente Nutzung der Batterielaufzeit, da die Hauptberechnungen auf dem Haupt ESP32 durchgeführt werden.

Die gesammelten Daten werden dann auf einer SD-Karte gespeichert, die am Netzwerkhub angeschlossen ist. Dadurch wird sichergestellt, dass die Daten sicher abgelegt werden. Ebenfalls können die Daten live über eine Web-App ausgelesen werden. Besondere Auffälligkeiten in den Daten werden markiert, um sofortiges Handeln zu ermöglichen.

Die einzelne Module der Sensorbox können auch kabelgebunden sein, falls eine längere Akkulaufzeit benötigt wird.

## Energiekonzept

#### **Periphere Sensorboxen:**

* **Sleep-Modus**: Die peripheren Boxen gehen für **5 Minuten** in den **Deep Sleep-Modus**, in dem der Stromverbrauch stark reduziert wird.
* **Aktivierungsphase**: Nach den 5 Minuten erwachen die Boxen, testen für **2 Minuten** die Sensoren und senden die Daten an die Main Sensorbox.
* **Zyklus**: Dieser Zyklus (5 Minuten Schlaf, 2 Minuten Aktivität) wird kontinuierlich wiederholt, um die **Akkulaufzeit** zu maximieren und den Energieverbrauch gering zu halten.

#### **Main Sensorbox:**

* **Dauerbetrieb**: Die Main Sensorbox läuft **permanent** und empfängt die Daten von den peripheren Boxen. Sie ist mit einer kontinuierlichen Stromversorgung (z. B. Netzteil) ausgestattet, um eine **dauerhafte Verfügbarkeit** sicherzustellen.

## Software

### **Programmiersprache:**

Da es sich hier um einen ESP32 handelt, ist hier sinnvoll Python zu benutzen.

* Vorteile
* Einfach
* Simpel
* Viele Bibliotheken
* Ähnliche Projekte sind in der selben Sprache geschrieben
* Nachteile
* Langsam
* Hoher Speicherverbrauch, wenn größere Daten geschickt werden.
* Ist keine Scriptsprache, wie z.B. C#

### **WLAN:**

* Da Daten geschickt werden, ist das Kommunikationsprotokoll vom WLAN einfach strukturiert und gut geeignet für den ESP32
* Zudem gibt es viele Vergleiche zum WLAN-Protokoll, da dies öfters bei Projekten benutzt werden

### **Sensor Daten ablesen:**

* Da Daten geschickt werden, müssen die Sensordaten auch erfasst werden
* Mithilfe von den Bibliotheken von Python kann die Erfassung vereinfacht werden
* Je nach Sensor, können die Daten unterschiedlich erfasst werden

### **Mesh-Netz:**

* Beim Mesh-Netz werden Daten per WLAN Protokoll zum Main Board weitergegeben
* Die Kommunikationsvermittlung kann mindestens 10m – 15m betragen (Bei WLAN Protokoll)
* Es können ausreichend Daten vermittelt werden

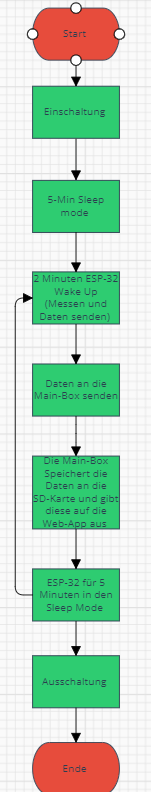
### **Website:**

* Für die Programmierung wird JavaScript benutzt
* Für die Darstellung wird HTML und CSS benutzt, da diese fast immer für Websites benutzt werden
* Die Website benutzt die Daten des ESP32, welche bei der SD-Karte gespeichert werden
* Mithilfe dieser Daten hat die Website die Möglichkeit, die Daten zu benutzen und sinnvoll in einer Graphischen Sicht darzustellen

### **Funktion der App:**

* Die App stellt die Website in einem schönen Format dar
* Die Daten für die App werden vom HUB-ESP geliefert
* Die App, z.B. beim Handy verbindet sich dann mit dem HUB-ESP, um die Daten zu kriegen
* Die Daten werden dann in der App, in einem schönen Diagramm dargestellt
* Beim Diagramm kann man auswählen, was angezeigt werden sollte, z.B. Temperatur oder Spannung

## Ablaufdiagramm



## Skizze zur Box

## Ein Bild, das Design enthält. Automatisch generierte Beschreibung

Platine

Sensor

Box

## Technische Daten

|  |  |  |
| --- | --- | --- |
| **Bezeichnung** | **Spezifikation** | **technischer Check Prototypen** |
| Eingangsspannung | 5V Akku |  |
| Betriebsbereich | -20°C bis 60°C |  |
| Netzgebunden | 100 Stunden |  |
| Netzunabhängig | 50 Stunden mit 3000mAh |  |
| Frequenz der aufgenommenen Daten | Alle 5 Minuten für 2 Minuten |  |
| Mindestgröße für die Main Box |  |  |
| Max Größe für die Sensorboxen | 5x10,5x3,5 cm groß |  |
| Batterie für Main Box | 5000 mAh |  |
| Batterie für Sensorboxen | 3000 mAh |  |
| Mesh-Netzwerk | WLAN |  |
| SanDisk SD |  |  |
| Temperatursensor-PT100 | Die Leitungen müssen temperaturbeständig sein, insbesondere in Umgebungen mit hohen oder sehr niedrigen Temperaturen. |  |
| Luftdrucksensor | Bei präzisen Messungen können abgeschirmte Leitungen erforderlich sein, um Rauschen zu minimieren. |  |
| Feuchtigkeitssensoren | Die Leitungen sollten resistent gegen Feuchtigkeit sein, um Korrosion zu vermeiden. |  |
| Ozonmesssensoren | Abgeschirmte Kabel, um elektromagnetische Störungen zu vermeiden. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Technischer Check Prototypen durchgeführt** | |
| siehe Freigabeblatt |  |

## Betriebsmodi / User Interface

|  |  |  |
| --- | --- | --- |
| **Bezeichnung** | **Spezifikation / Auflösung / Anzahl** | **Implementierung Prototypen** |
| **Modus** |  |  |
| Batteriemodus | 50 Stunden |  |
| Kabelgebundener Modus | Min. 100 Stunden |  |
|  |  |  |
| **Anzeige** |  |  |
| Mittels SD – Karte speichern |  |  |
| Anzeige auf den PC mittels Web-App |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Check Implementierung Prototypen (Betriebsmodi) durchgeführt** | |
| siehe Freigabeblatt |  |

## PC/Host-Applikation

|  |  |  |
| --- | --- | --- |
| **Bezeichnung** | **Spezifikation / Auflösung / Anzahl** | **Implementierung Prototypen** |
| **Netzwerkhub** | ESP32-basierter Hub, 5 Geräte |  |
| **Datenspeicherung** | SD-Kartenmodul, 32GB |  |
| **Datenvisualisierung** | Web-App, Javascript |  |
| **Datenanalyse** | Echtzeitverarbeitung |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Check Implementierung Prototypen (PC/Host-Applikation) durchgeführt** | |
| siehe Freigabeblatt |  |

## Normen

**Sicherheit**

EN 61010-1 Sicherheitsbestimmungen für elektrische  
Mess-, Steuer-, Regel- und Laborgeräte

**EMV Immunität**

EN 61000-4-2 Elektrostatische Entladung

**EMV Emmission**

EN 61000-6-4 Fachgrundnorm - Störaussendung für Industriebereiche

|  |  |  |
| --- | --- | --- |
| **Norm / Prüfgegenstand** | **Dokumentation** | **Prüfergebnis** |
| - | - | - |
| - | - | - |
| - | - | - |
| **-** | - | - |

|  |  |
| --- | --- |
| **Technischer Check Prototypen (Normen) durchgeführt** | |
| siehe Freigabeblatt |  |

# Projektkosten

## Materialkosten

Die genauen Materialkosten können derzeit noch nicht vollständig bestimmt werden. Wir schätzen jedoch, dass die Kosten für die benötigten Materialien zwischen 200 und 300 Euro liegen werden. Dies umfasst Komponenten wie den ESP32, SD-Kartenmodule und andere notwendige Hardware.

## Sonstige Kosten

Auch diese Kosten können aktuell nicht bestimmt werden. Wir erwarten externe Kosten von etwa 100 Euro, die beispielsweise für Softwarelizenzen, spezielle Werkzeuge oder sonstige Ausgaben anfallen könnten.

# Arbeitsplan / Meilensteine

## Aufteilung in Hardware und Software:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unterprojekt** | **Pr.Nummer** | **Aufwand (h)** | **Termin  MM/JJ** | **Arbeitsschritte** |
| Anforderungsanalyse | - | Ca. 20-30 |  |  |
| Hardwareentwicklung | - | Ca. 90 |  |  |
| Prototyping Hardware | - | Ca. 30 |  |  |
| Softwareentwicklung | - | Ca. 150 |  |  |
| Tests und Validierung | - | Ca. 30 |  |  |
| Dokumentation | - | 300 |  |  |
| Präsentation | - | 10 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Design-Verifizierung

## Vor Freigabe der Prototypenfertigung werden zur Kontrolle die folgenden technischen Dokumente überprüft (sofern relevant).

## Technische Unterlagen Elektronik

O Schaltpläne erstellt und geprüft

O Layouts erstellt und geprüft

O Stücklisten erstellt und geprüft

O Prüfanweisungen erstellt und geprüft, Prüfmittel definiert

O Fertigungsunterlagen erstellt und Fertigungsspezifikation geprüft

## Technische Unterlagen Mechanik

O Teilezeichnungen erstellt und geprüft

O Gehäusezeichnungen erstellt und geprüft

O Konstruktionsdaten (3D) erstellt und geprüft

O Wareneingangsprüfung erstellt und geprüft

## Technische Unterlagen Software / Firmware

O Prüfprogramme erstellt und getestet

O Hardware-Steuerungsteil erstellt und getestet

O User-Interface erstellt und getestet

O PC-Software erstellt und getestet

## Spezifikationen / Administrative Unterlagen

O Entwicklung entspricht voraussichtlich den technischen Spezifikationen

O Entwicklung entspricht voraussichtlich den geforderten Normen

O Artikel für Bestellwesen definiert und angelegt

|  |  |
| --- | --- |
| **Verfizierung durchgeführt** | **Bemerkung / Maßnahmen** |
|  |  |
|  |  |
|  |  |
|  |  |

# Prototypen / Serienfreigabe

## Freigabe Fertigung Prototypen

|  |  |
| --- | --- |
| **Designverifizierung erfolgreich, Freigabe für Prototypenfertigung (F5)** | |
| siehe Freigabeblatt |  |

## Überprüfung Prototypen

* technische Daten überprüfen (siehe Checkliste technische Daten)
* Herstellkosten aktualisieren (dokumentiert in Tabelle Herstellkosten)
* Gefahrenanalyse/Risikoanalyse aktualisieren

## Checkliste für den Fertigungsstart

O technische Unterlagen auf Stand

O Gefahrenanalyse erstellt

O Artikel für Bestellwesen definiert

O Prüfanweisungen aktualisiert

O Wareneingangsprüfungen aktualisiert

O Aufbauanleitungen erstellt

O Geräte-Endprüfung definiert

O Kalibrieranweisung und Zertifikat erstellt

O Geräte-Datenblatt erstellt

O Bedienungsanleitung erstellt

O Verpackungskonzept erstellt

|  |  |
| --- | --- |
| **Durchgeführt von** | **Bemerkung / Maßnahmen** |
|  |  |
|  |  |
|  |  |
|  |  |

## Freigabe Serienfertigung

|  |  |
| --- | --- |
| **Produkt technisch und wirtschaftlich herstellbar, Freigabe für Serienfertigung (F6)** | |
| siehe Freigabeblatt |  |

# Änderungsvermerke Projekt / Serienstand

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Name** | **Betroffene Baugruppen** | **Änderungsgrund / -beschreibung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Kennzahlen

|  |  |  |
| --- | --- | --- |
| **L. Kennzahlen** | | |
| **Kennzahl** | **Bewertung** | **Bemerkung** |
| Einhaltung des EW-Budgets |  | 2: ja / 1:im Rahmen / 0:nein |
| Einhaltung der EW-Termine |  | 2: ja / 1:im Rahmen / 0:nein |
| Qualität der Entwicklung |  | 2: gut / 1:akzeptabel /  0: nicht ausreichend |
| **Kennzahlen erstellt:** | | |

# Gesprächsprotokolle

Datum: Thema: **Kick-Off Meeting**

Teilnehmer:

|  |  |  |
| --- | --- | --- |
| Besprechungspunkte | Maßnahmen | Umsetzung bis |
|  |  |  |
|  |  |  |
|  |  |  |

Datum: Thema:

Teilnehmer:

|  |  |  |
| --- | --- | --- |
| Besprechungspunkte | Maßnahmen | Umsetzung bis |
|  |  |  |
|  |  |  |
|  |  |  |

Datum: Thema:

Teilnehmer:

|  |  |  |
| --- | --- | --- |
| Besprechungspunkte | Maßnahmen | Umsetzung bis |
|  |  |  |
|  |  |  |
|  |  |  |