# Handout Lernfeld 7

# 2.1 Grundlagen von cyber-physischen Systemen erarbeiten

## 2.1.1 Was sind cyber-physische Systeme?

### Definition:

* + Cyber-physische Systeme (CPS) verbinden physische Komponenten mit digitalen Technologien.
  + Sie integrieren Sensoren, Aktoren und Software, um physikalische Prozesse zu überwachen und zu steuern.

### Eigenschaften:

Echtzeitfähigkeit:

* + - Reagieren unmittelbar auf Änderungen in der Umgebung.

Konnektivität:

* + - Vernetzung durch Protokolle wie MQTT oder OPC UA.

Autonomie:

* + - Treffen Entscheidungen basierend auf Daten und Algorithmen.

Skalierbarkeit:

* + - Systeme können flexibel erweitert oder reduziert werden.
* Beispiele:
  + Smart Home: Automatisierung von Heizung, Beleuchtung und Sicherheit.
  + Industrie 4.0: Produktionsanlagen, die durch Sensoren überwacht und optimiert werden.
  + Medizin**:** Intelligente Geräte zur Patientenüberwachung.

## 2.1.2 Ziele und Nutzen von CPS

* Ziele:
  1. Verbesserung der Effizienz und Qualität von Prozessen.
  2. Automatisierung und Optimierung durch intelligente Steuerung.
  3. Integration physischer und digitaler Komponenten für neue Funktionalitäten.
* Nutzen:
  1. Für Unternehmen:
     + Kosteneinsparungen, Produktivitätssteigerung, innovative Geschäftsmodelle.
  2. Für Endnutzer:
     + Komfort, Sicherheit und personalisierte Lösungen.

### 2.1.3 Bestandteile eines CPS

1. Sensoren:
   * Erfassen physikalische Größen (z. B. Temperatur, Feuchtigkeit).
   * Beispiele: CO2-Sensoren, Bewegungssensoren.
2. Aktoren:
   * Setzen digitale Signale in physische Aktionen um.
   * Beispiele: Motoren, Ventile.
3. Mikrocontroller:
   * Zentraler Steuerungsknoten, verarbeitet Sensordaten und steuert Aktoren.
   * Beispiele: Arduino, ESP32.
4. Kommunikationsschnittstellen:
   * Protokolle und Netzwerke für die Datenübertragung.
   * Beispiele: WLAN, MQTT, OPC UA.
5. Software:
   * Algorithmen und Programme für Steuerung und Analyse.
   * Dashboards zur Visualisierung der Ergebnisse.

### 2.1.4 Herausforderungen bei der Einführung von CPS

1. Sicherheitsrisiken:
   * Gefahr durch unbefugten Zugriff oder Datenmanipulation.
   * Lösung: Verschlüsselung und sichere Authentifizierung.
2. Kosten und Komplexität:
   * Hohe Anfangsinvestitionen und technologische Integration.
   * Lösung: Schrittweise Einführung und Nutzung standardisierter Komponenten.
3. Kompatibilität:
   * Unterschiedliche Standards und Protokolle erschweren die Integration.
   * Lösung: Einsatz universeller Kommunikationsschnittstellen.
4. Schulung und Akzeptanz:
   * Mitarbeiter müssen die neuen Systeme verstehen und nutzen können.
   * Lösung: Schulungen und Workshops.
5. Funktionen von CPS

### Begriff Definition

Cyber-physische Produktionssysteme (CPPS) Einsatz von CPS in der Industrie zur intelligenten Steuerung und Überwachung von Produktionsanlagen.

Internet of Things (IoT) Globale Infrastruktur, die physische und virtuelle Objekte über das Internet verbindet.

Industrial Internet of Things (IIoT) Anwendung des IoT in industriellen Prozessen; intelligente Sensoren vernetzen Maschinen und Anlagen.

Smart Factory Produktionsumgebung, die sich selbst organisiert und optimiert; basiert auf vernetzten Maschinen und cyber-physischen Systemen.

## Internet of Things (IoT)

* Definition:  
  Vernetzung von physischen und virtuellen Objekten über das Internet.

### Technische Grundlagen:

* + Konnektivität: Verschiedene Systeme, Standards und Schnittstellen müssen miteinander kommunizieren können (z. B. 4G/5G, WLAN, OPC UA).
  + Offene Schnittstellen: APIs ermöglichen universellen Zugriff auf IoT-Plattformen.
  + Datenanalyse: Echtzeitüberwachung und -analyse zur Prozessoptimierung.
  + Sicherheit: Authentifizierung, Verschlüsselung und sicherer Datentransfer.

## 2.2 Kommunikation zwischen cyber-physischen Systemen präsentieren

### 2.2.1 Grundlagen der Kommunikation zwischen CPS

* Definition:
  + Kommunikation zwischen CPS beschreibt den Datenaustausch zwischen Sensoren, Aktoren und anderen Systemkomponenten, um Informationen in Echtzeit auszutauschen.
* Anforderungen an die Kommunikation:
  + Zuverlässigkeit: Datenübertragung ohne Fehler oder Verluste.
  + Echtzeitfähigkeit: Reaktion in Millisekunden oder Sekunden.
  + Skalierbarkeit: Unterstützung vieler Geräte.
  + Sicherheit: Schutz vor unbefugtem Zugriff und Datenmanipulation.
* Typen der Kommunikation:
  + Gerät-zu-Gerät (D2D):
    - Direkte Kommunikation zwischen CPS-Komponenten, z. B. Sensoren und Aktoren.
  + Gerät-zu-Cloud:
    - Daten werden an eine zentrale Plattform zur Verarbeitung und Analyse gesendet.
  + Gerät-zu-Mensch:
    - Interaktion über Dashboards oder Apps.

### 2.2.2 Kommunikationsprotokolle

* Wichtige Protokolle:
  1. MQTT (Message Queuing Telemetry Transport):
     + Leichtgewichtiges Protokoll für den Datenaustausch.
     + Eignet sich besonders für IoT- und CPS-Anwendungen.
     + Komponenten:
       - Broker: Vermittelt Nachrichten.
       - Publisher/Subscriber: Sendet/empfängt Daten zu bestimmten Themen („Topics“).
  2. OPC UA (Open Platform Communications Unified Architecture):
     + Plattformunabhängiges Protokoll für industrielle Anwendungen.
     + Vorteile:
       - Unterstützt Sicherheitsfeatures (z. B. Verschlüsselung).
       - Ermöglicht semantische Datenbeschreibung.
  3. HTTP/HTTPS:
     + Standardprotokoll für Web-Kommunikation.
     + Sicher (HTTPS) und weit verbreitet, aber nicht optimal für Echtzeitanforderungen.
  4. CoAP (Constrained Application Protocol):
     + Speziell für ressourcenbeschränkte Geräte entwickelt.
     + Nutzt UDP (statt TCP) für geringere Latenzen.
* Vergleich der Protokolle:

| Protokoll | Einsatzgebiet | Vorteile | Nachteile |
| --- | --- | --- | --- |
| MQTT | IoT, CPS | Leichtgewichtig, effizient | Broker notwendig |
| OPC UA | Industrie, Maschinensteuerung | Sicherheit, plattformunabhängig | Höherer Ressourcenbedarf |
| HTTP/HTTPS | Web-Dashboards | Universell, sicher | Hohe Latenz |
| CoAP | Ressourcenschwache Geräte | Geringe Latenz, effizient | Weniger verbreitet |

2.2.3 Netzwerktechnologien

* Verbindungsmethoden:
  1. Kabelgebundene Netzwerke:
     + Ethernet: Stabil, hohe Übertragungsgeschwindigkeiten, ideal für industrielle CPS.
  2. Drahtlose Netzwerke:
     + WLAN (Wi-Fi): Geeignet für flexible Anwendungen, aber potenziell störanfällig.
     + Bluetooth Low Energy (BLE): Für energieeffiziente Kurzstreckenkommunikation.
     + LoRaWAN: Für Langstreckenkommunikation mit geringem Energieverbrauch.
* Topologien:
  1. Stern-Topologie:
     + Zentrale Einheit (z. B. Router) verbindet alle Geräte.
     + Vorteile: Einfache Fehleridentifikation.
     + Nachteile: Abhängigkeit von der Zentrale.
  2. Mesh-Topologie:
     + Geräte kommunizieren direkt miteinander.
     + Vorteile: Hohe Ausfallsicherheit.
     + Nachteile: Komplexere Verwaltung.

2.2.4 Herausforderungen der CPS-Kommunikation

1. Latenzzeiten minimieren:
   * Verzögerungen in der Datenübertragung beeinflussen die Systemleistung.
   * Lösung: Einsatz von Echtzeitprotokollen wie MQTT oder OPC UA.
2. Skalierung bei steigender Geräteanzahl:
   * Mehr Geräte führen zu höherem Datenaufkommen.
   * Lösung: Lastverteilung durch Broker und Cloud-Lösungen.
3. Sicherheitsrisiken:
   * Gefährdung durch Hackerangriffe oder Datenverlust.
   * Lösung: Verschlüsselung, Authentifizierung und regelmäßige Sicherheitsupdates.
4. Interoperabilität:
   * Verschiedene Hersteller verwenden unterschiedliche Standards.
   * Lösung: Einsatz universeller Protokolle wie OPC UA.

## 2.3 Ein bestehendes cyber-physisches System im Unternehmen analysieren

### 2.3.1 Ziel und Zweck der Analyse

* Ziel:
  + Evaluierung der Funktionsfähigkeit, Effizienz und Optimierungspotenziale eines bestehenden CPS.
  + Sicherstellung, dass das System den aktuellen Anforderungen gerecht wird.
* Bedeutung der Analyse:
  + Fehler erkennen: Identifikation von Schwachstellen im System.
  + Optimierungspotenziale: Verbesserung von Leistung, Sicherheit und Interoperabilität.
  + Langfristige Planung: Vorbereitung auf zukünftige Anforderungen und Technologien.

### 2.3.2 Vorgehensweise bei der Analyse

1. Systemaufnahme und Dokumentation

* Systemkomponenten identifizieren:
  + Sensoren: Erfassen physikalische Werte (z. B. Temperatur, Feuchtigkeit).
  + Aktoren: Steuern physikalische Prozesse (z. B. Ventile, Motoren).
  + Mikrocontroller/Edge-Geräte: Verarbeiten Daten und steuern Geräte.
  + Kommunikationsschnittstellen: Ermöglichen den Datenaustausch.
  + Dashboards: Visualisieren Daten für Nutzer.
* Systemkontext definieren:
  + Abgrenzung der Systemgrenzen (Was gehört zum System?).
  + Relevante Umgebung (z. B. Netzwerke, physikalische Einflüsse).
* Bestandsdokumentation:
  + Erstellung von Plänen und Übersichten:
    - Topologien der Kommunikation.
    - Verwendete Protokolle (z. B. MQTT, OPC UA).
    - Systemarchitektur (zentral vs. dezentral).

2. Funktionsanalyse

* Prüfung der Funktionalität:
  + Erfassen, ob Sensoren, Aktoren und Steuerungen wie vorgesehen arbeiten.
  + Überprüfen, ob die Daten korrekt verarbeitet und übertragen werden.
* Anwendungsfälle überprüfen:
  + Erfüllung der Hauptaufgaben (z. B. Klimaüberwachung, automatisierte Steuerung).
  + Auswertung der Nutzerfreundlichkeit und Effizienz des Dashboards.
* Leistungsbewertung:
  + Geschwindigkeit der Datenverarbeitung und -übertragung.
  + Verfügbarkeit und Stabilität des Systems.

3. Fehler- und Schwachstellenanalyse

* Typische Probleme:
  + Kommunikationsabbrüche: z. B. durch Netzwerkfehler.
  + Sensorfehler: Falsche oder keine Werte werden erfasst.
  + Sicherheitslücken: Unzureichender Schutz vor Cyberangriffen.
* Diagnosetools:
  + Logs und Protokolldaten analysieren.
  + Testgeräte zur Simulation und Fehlersuche einsetzen.
* Bewertung der Fehlerauswirkungen:
  + Kritische Fehler: Beeinträchtigen wesentliche Funktionen.
  + Geringfügige Fehler: Beeinflussen das System nur minimal.

4. Bewertung der Betriebs- und Datensicherheit

* Aspekte der Sicherheit:
  + Datenintegrität: Schutz vor Manipulation der Daten.
  + Zugriffsschutz: Nur autorisierte Nutzer können das System bedienen.
  + Verschlüsselung: Schutz der Datenübertragung (z. B. TLS).
* Sicherheitsmaßnahmen prüfen:
  + Rollenbasierte Zugriffskontrollen (Role-Based Access Control).
  + Einsatz von Firewalls und Antivirus-Programmen.
  + Regelmäßige Updates und Patches.

5. Identifizierung von Optimierungspotenzialen

* Interoperabilität verbessern:
  + Standardisierung der Protokolle und Schnittstellen.
  + Einsatz von universellen Plattformen wie OPC UA.
* Leistungsfähigkeit steigern:
  + Einsatz leistungsfähigerer Hardware (z. B. Mikrocontroller mit höherer Rechenleistung).
  + Optimierung von Datenverarbeitungs- und Übertragungsalgorithmen.
* Flexibilität erhöhen:
  + Erweiterung des Systems für neue Sensoren oder Aktoren.
  + Cloud-Integration zur Analyse großer Datenmengen.

2.3.3 Ergebnisse und Dokumentation

* Erstellte Berichte:
  1. Systemübersicht: Komponenten, Funktionen und Schnittstellen.
  2. Fehler- und Schwachstellenbericht: Identifizierte Probleme und Ursachen.
  3. Optimierungsvorschläge: Konkrete Maßnahmen zur Verbesserung.
* Methoden der Ergebnisdokumentation:
  1. Diagramme (z. B. Funktionsdiagramme, Kommunikationswege).
  2. Tabellen für Fehlerkategorien und Prioritäten.
  3. Nutzung von Tools wie Fritzing zur Visualisierung elektronischer Schaltungen.
* Kommunikation der Ergebnisse:
  1. Bericht an Stakeholder und Entscheidungsträger.
  2. Diskussion und Abstimmung der Optimierungsvorschläge.

2.3.4 Herausforderungen bei der Analyse

1. Komplexität des Systems:
   * Vielzahl von Geräten und Protokollen erschwert die Analyse.
2. Datenmenge:
   * Hoher Aufwand für die Auswertung großer Datenvolumen.
3. Sicherheitsbedenken:
   * Analyse darf keine bestehenden Sicherheitsrisiken verstärken (z. B. durch Testsysteme).
4. Interdisziplinarität:
   * Zusammenarbeit verschiedener Fachbereiche (z. B. IT, Fertigung).

## 2.4 Erweiterungen für ein cyber-physisches System planen

### Projektaufgabe:

* Ausstattung von Arbeitsräumen mit Sensoren zur Erfassung von:
  + Raumtemperatur
  + Luftfeuchtigkeit
  + CO₂-Konzentration
* Automatisierte Steuerung des Raumklimas und der Belüftung.

### Anforderungen:

* Darstellung von Informationen auf einem Dashboard.
* Einbindung der CPS-Devices ins Firmennetzwerk.
* VUCA-Herausforderungen:
  + Volatility: Schnelle Veränderungen erfordern klare Visionen.
  + Uncertainty: Unklare Vorhersagen erfordern Verständnis von Zusammenhängen.
  + Complexity: Komplexität verlangt Klarheit und einfache Schritte.
  + Ambiguity: Mehrdeutigkeit erfordert Anpassungsfähigkeit.

### 2.4.1 Anforderungsanalyse durchführen

#### Requirements Engineering:

* Ziele:
  + Systematische Spezifikation und Verwaltung von Anforderungen.
  + Klare Dokumentation (Lasten- und Pflichtenheft).
* Wichtige Begriffe:
  + Anforderungskatalog: Sammlung von Kundenanforderungen.
  + Anwendungsszenarien: Beschreiben Nutzerinteraktionen (z. B. „CO₂-Werte erfassen“).
  + Systemkontext: Definiert die Interaktion zwischen System und Umgebung.

#### Stakeholder:

* Berücksichtigen Sie Interessen von:
  + Endkunden
  + Mitarbeitern
  + Institutionen, Behörden, Gesetzgebern.

#### Akzeptanzkriterien:

* Anforderungen müssen klar und überprüfbar sein.
* Vorteile:
  + Höhere Transparenz.
  + Klare Grundlage für Tests und Stakeholder-Kommunikation.

### 2.4.2 Projektplanung mit agilen Methoden

#### Agiles Projektmanagement mit DevOps:

* Ziele:
  + Höhere Bereitstellungsfrequenz.
  + Reduzierung von Fehlern.
  + Kürzere Markteinführungszeit.
* Phasen:
  + Planen: Anforderungen festlegen.
  + Codieren: Programmieren mit Versionskontrolle (z. B. Git).
  + Testen: Automatisierte Tests durchführen.
  + Bereitstellen: Produktionsfreigabe der Software.
  + Überwachen: Kontinuierliche Verbesserung durch Feedback.

#### CALMS-Modell:

* Culture: Zusammenarbeit und Kundenorientierung.
* Automation: Effizienz durch Automatisierung.
* Lean: Fokus auf Mehrwert.
* Measurement: Datengestützte Entscheidungen.
* Sharing: Kollaborativer Wissensaustausch.

## 2.5 Umsetzung eines CPS-Prototyps

### 2.5.1 Prototyp entwickeln:

* Ziel: Aufbau eines funktionsfähigen Modells mit:
  + Arduino- und ESP32-Mikrocontroller.
  + Sensoren und Aktoren.
* Dokumentation: Mit Tools wie „Fritzing“ elektrische Schaltungen visualisieren.

### 2.5.2 Softwareentwicklung:

* + Drei bereiche innerhalb eines Sketchs

### 2.5.3 MQTT-Broker einrichten:

-

### 2.5.4 Funktionalität prüfen:

* Funktionstests:
  + Validierung der Sensorwerte.
  + Debugging mit seriellen Monitoren.
* Dokumentation: Protokollierung aller Ergebnisse.

### 2.5.5 Lessons Learned:

* Ziel: Erfahrungen systematisch erfassen und analysieren.
* Fragestellungen:
  + Was lief gut/schlecht?
  + Was kann verbessert werden?
  + Welche Erkenntnisse sind für künftige Projekte relevant?

## 2.6 Optimierung eines CPS

### 2.6.1 Sicherheitsrisiken erkennen:

* Herausforderungen:
  + Bedrohungen durch IT/OT-Konvergenz.
  + Potenzielle Risiken durch Angriffe oder Ausfälle.
* Maßnahmen:
  + Kritische Prozesse identifizieren.
  + Mobile Geräte sichern (z. B. Mobile-Device-Management).
  + Zugriffsberechtigungen überwachen.

### Systemausfälle mit FMEA minimieren:

* Ablauf FMEA (7 Schritte)
* RPZ**-**Berechnung**:** RPZ=B×A×ERPZ = B \times A \times ERPZ=B×A×E
  1. B: Bedeutung der Fehlerfolge.
  2. A: Auftretenswahrscheinlichkeit.
  3. E: Entdeckungswahrscheinlichkeit.

### 2.6.3 Verbesserungsvorschläge präsentieren:

* Methoden:
  1. Betriebliches Vorschlagswesen: Systematische Einreichung von Ideen.
  2. Elevator Pitch: Kurzpräsentation nach AIDA-Modell:
     + Attention: Aufmerksamkeit gewinnen.
     + Interest: Interesse wecken.
     + Desire: Verlangen nach Umsetzung erzeugen.
     + Action: Handlungsaufruf.