Cyber Physical Systems

Wintersemester 2020/21

Hochschule München University of Applied Sciences

Trainingszeitmesssystem für Oldtimer Gleichmäßigkeitsrallyes

von: Max Holzmann

Diese Tutorial Reihe erklärt detailliert die Implementierung eines kleinen Cyber Physical Systems (CPS). Bei dem CPS handelt es sich um eine Zeitmessanlage welche beim Training für Oldtimer Gleichmäßigkeitsrallyes zum Einsatz kommen kann.

Ziele und verwendete Technologien

- Messanlage mit Lichtschranke
    - Berechnung der Zeitdifferenz zwischen der nächsten vollen Minute und der Zeit der Durfahrt mit dem Oldtimer durch die Lichtschranke
    - Grundgedanke:
        - Ein Raspberry Pi als RallyMaster und Basis des Systems
        - Mehrere ESP32 als sogenannte RallyESPs in den Oldtimern zur Visualisierung (visuell und akustisch) der Ergebnisse
- Verwendete Technologien
    - **4G Mobilfunk** zur globalden Zeitsynchronisation
    - **Wifi Accesspoint** zur lokalen Kommunikation zwischen Rally Master und Rally ESPs
    - **Node Red** als Hauptanwendung zur Technologie Verknüpfung

    - **MQTT** als Netzwerk Übetragungsprotokoll
    - **NFC** zur RallyESP registrierung beim RallyMaster
    - **MongoDB** als Datenbank für die Messergebnisse
    - **OPC-UA** als Schnittstelle zur Abfrage von RallyMaster Werten
    - **E-Paper** Display zur Darstellung der Ergebnisse am RallyESP

Vorraussetzungen und erforderliches Equipment

- Hardware
    - RallyMaster
        - 1x RaspberrPi 3 B shopping link
        - 1x 7" Raspberry Touch Display (optional) shopping link
        - 2x NFC Board mit PN532 shopping link
        - 1x ESP DevKitC shopping link
        - 4G Raspberry Pi HAT mit SIM7600E + SIM Karte shopping link
        - Lichtschranke
            - Eigenbau Industrie Reflexlichtschranke auf Stativ mit Optokopplerschaltung zu Anbindung an die GPIO Schnittstelle des Raspberry Pi
        - Kosten: ca. 200€
    - pro RallyESP
        - 1x ESP DevKitC shopping link
        - 1x NFC Board mit PN532 shopping link
        - 1x Waveshare E-paper 2,9" shopping link
        - Kosten ca 40€
- Verwendete Software und verwendete Tools in diesem Tutorial
    - Entwiklungsumgebung für den ESP32
        - Arduino IDE
            - Version 1.8.13
        - Visual Studio Code
            - Windows 10
            - Visual Studio Code (VS Code)
            - VS Code Extention: Espressiv
    - Tools für den Raspberry Pi
        - Putty als SSH Terminal
        - VNC Viewer als remote desktop connection
        - win32DiskImager als OS immager

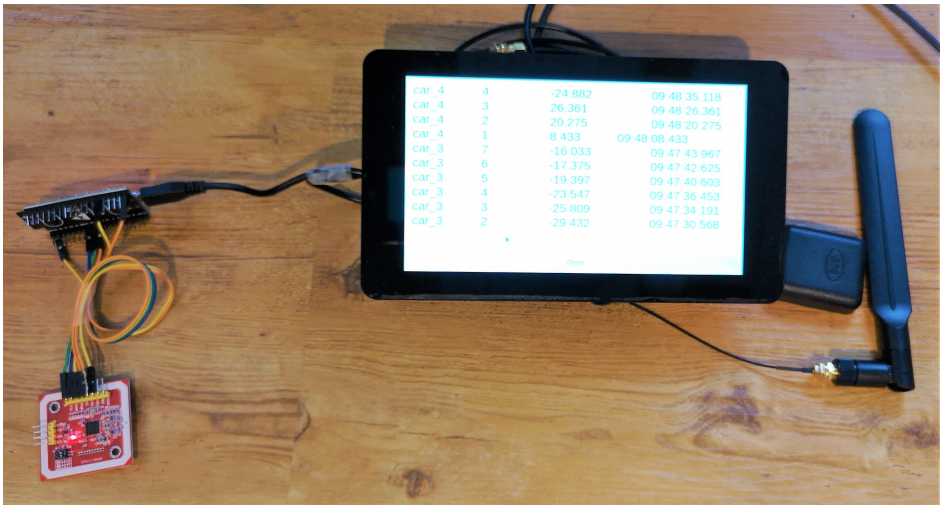
Beschreibung des gesamten Cyber Physical Systems

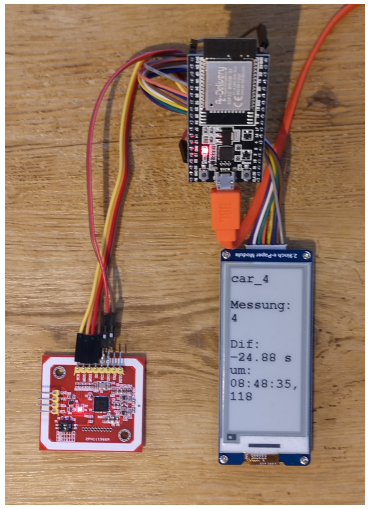
Mithilfe dieses CPS kann man die Zeitprüfungen von Oldtimerrallyes üben. Ziel einer Zeitprüfung ist es, möglichst genau bei der vorgegebenen Zeit die Lichtschanke mit dem Auto auszulösen.

Zu Trainigszwecken fährt man nun immer zu einer vollen Minute (DCF77 oder eines vorgegebenen NTP servers) durch die Lichtschranke. Das CPS wertet die Differenz zur vollen Minute in Millisekunden aus und

stellt sie dem Fahrer bereit. Das CPS ist so aufgebaut, dass es einen RallyMaster an der Lichtschranke gibt und mehrere RallyESPs in den teilnehmenden Oldtimern.

Der RallyMaster links und ein RallyESP rechts:
(Prototypen, im nächsten Entwicklungsschritt werden die Bauteile in einem 3D gedruckten Gehäuse untergebracht und um den Akkubetrieb erweitert):



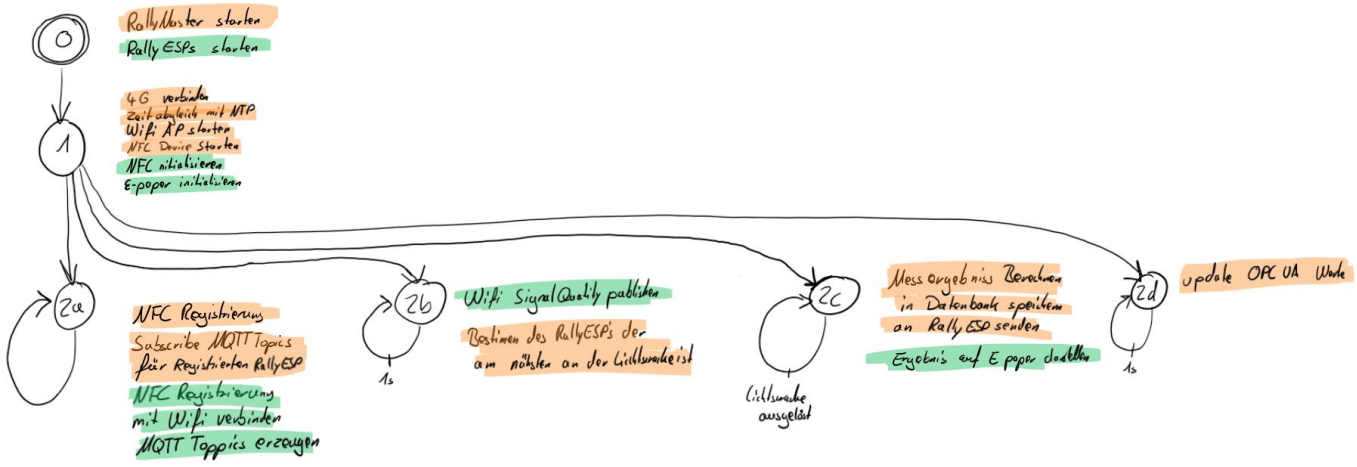


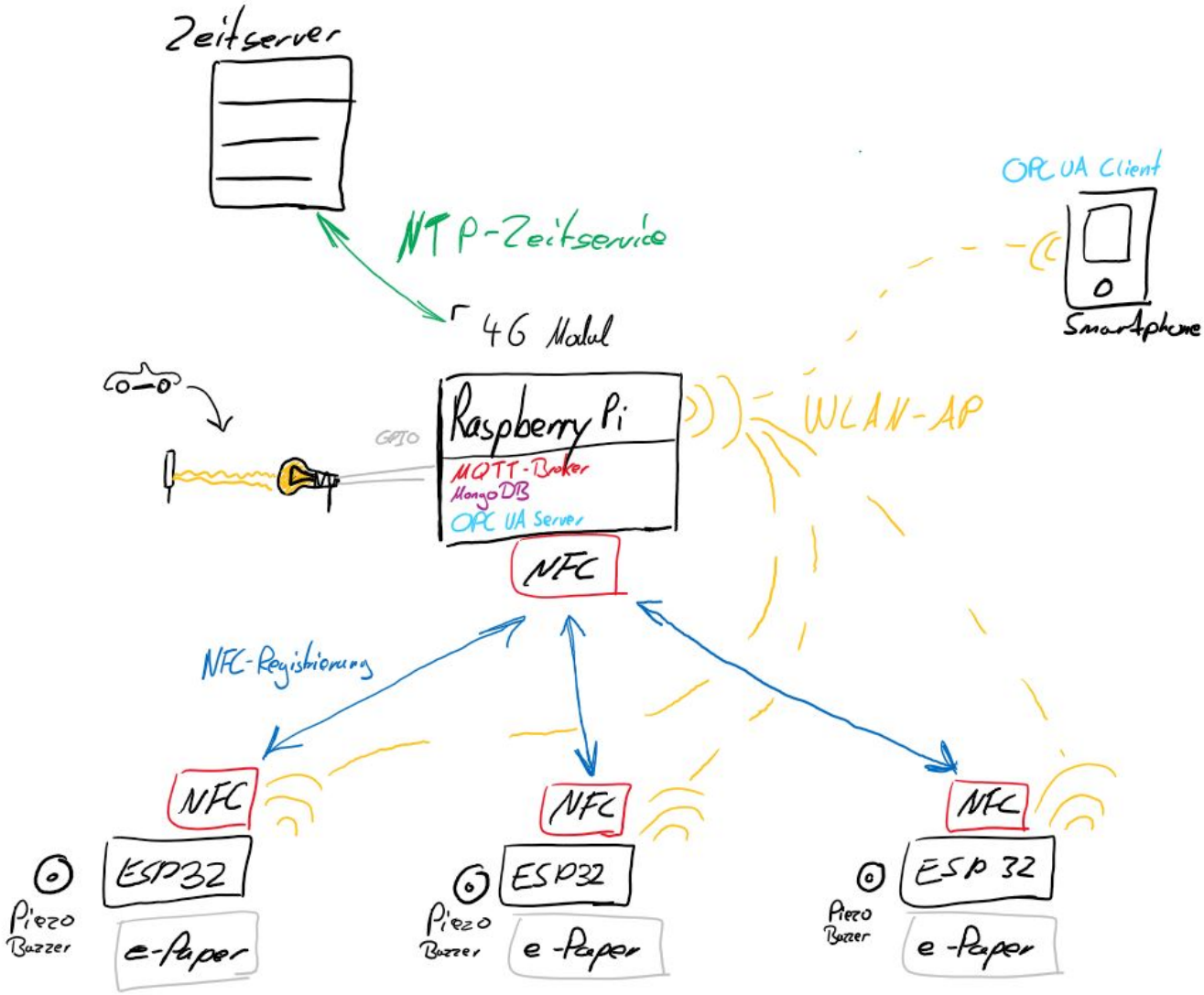
Möchte ein Teilnehmer trainieren, so meldet er sich mit seinem RallyESP via NFC beim RallyMaster an und erhält eine eindeutige Nummer. Anschließend verbindet sich der RallyESP mit dem WLAN welches der RallyMaster aufspannt. Über die WLAN Verbindung erfolgt nun die weitere Kommunikation.

Fährt der Teilnehmer mit seinem Oldtimer(incl. RallyESP) durch die Lichtschranke,wird ihm über MQTT sein Ergebnis zugespielt.

Wenn mehrere Teilnehmer und somit mehrere RallyESPs registriert sind, weiss der RallyMaster durch die Signalstärke der WLAN-Verbindung, welcher RallyESP gerade die Lichtschranke ausgelöst hat. So wird sichergestellt, dass das Ergebnis beim richtigen Teilnehmer angezeigt wird.

Der detaillierte Ablauf und das Blockdiagramm des CPS kann den folgenden Abbildungen entnommen werden.





Lösungsschritte

zwei Wege der Implementierung

1. Schritt für Schritt Anleitung zur Selbstimplementierung

    1. Lektion 01: Grundlegende Einstellungen der Entwicklungsumgebung und aufsetzen des Pi (link zur Lektion)
    2. Lektion 02: Raspberry mit dem 4G Netz verbinden (link zur Lektion)

3. Lektion 03: Konfigurieren des WLAN chips des Raspberry pi als Acesspoint (link zur Lektion)
4. Lektion 04: Node Red installieren (link zur Lektion)
5. Lektion 05: NFC Registrierung (link zur Lektion)
6. Lektion 06: MQTT Broker einrichten und in Rally Anwendung Integrieren (link zur Lektion)
7. Lektion 07: MongoDB einrichten und in Rally Anwendung Integrieren (link zur Lektion)
8. Lektion 08: OPC UA Server einrichten und mit Android APP testen (link zur Lektion)
9. Lektion 09: E-paper als Visualisierung der Ergebnisse am RallyESP (link zur Lektion)
10. Testen der Funktionen

2. Verwenden der fertigen Programme aus dieser Tutorial Reihe

1. Lektion 01: Grundlegende Einstellungen der Entwicklungsumgebung (aufsetzen des Pi nicht erforderlich)(link zur Lektion)
2. Fertiges Raspberry Pi 3 B Image der Tutorial Reihe aus dem Ordner 0110_Raspberry_Image mithilfe eines Imagers auf die SD Karte des Pi schreiben. Pi booten und ggf. Netzwerkeistellungen anpassen. Details können in den Lektionen der Tutorial Reihe nachgelesen werden.(link zur Lektion
3. RallyESP Programm aus dem Ordner 0100_Rally_ESP_Code in die Arduino IDE Laden, Übersetzen und auf einen ESP32 übertragen und testen.(link zur Lektion)
4. Testen der Funktionen