

# Projekt Jablotron – RS485 – HomeMatic<sup>IP</sup>

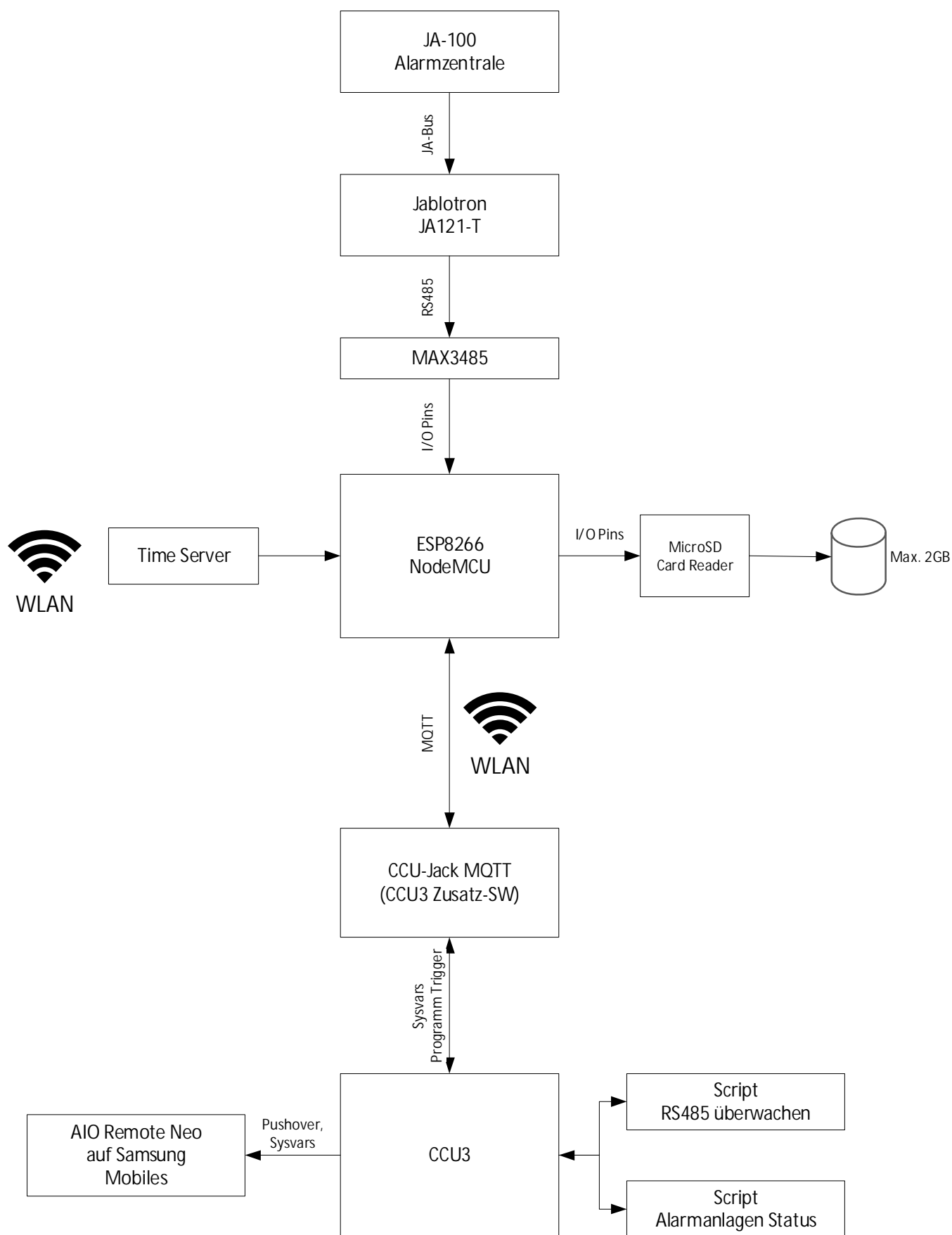
Dieses Dokument versteht sich nicht als Detailbeschreibung, sondern als Übersicht für die Zusammenhänge der einzelnen Komponenten und Referenz zu detaillierten Anleitungen bzw. als Quellenachweis.

**Projektziel:** Verfügbarkeit des Status der Jablotron Alarmanlage, sowie der Sensoren auf der HomeMatic<sup>IP</sup> CCU3.

## Komponenten:

- Jablotron 100
- Jablotron Modul JA-121T
- MAX3485
- Mikrokontroller ESP8266 (NodeMCU 1.0 / ESP12-F)
- MicroSD SPI Speicherkartenmodul
- Platine und Gehäuse für ESP8266 & MAX3485
- Sketch für ESP8266 (Arduino IDE)
- WLAN SSID
- CCU-Jack auf HomeMatic<sup>IP</sup> CCU3 installiert
- HomeMatic<sup>IP</sup> Scripts
- Pushover
- AIO Creator Neo

## Datenfluss Übersicht:



## Jablotron 100 und JA-121T:

Die JA-121T wird über die Busschnittstelle der Jablotron 100 angebunden. Diese sendet alle 10 Sekunden einen Heartbeat („OK“). Änderungen der Sensoren werden in der PRFSTATE „Bitleiste“ dargestellt. Der Status der Alarmanlage erfolgt über weitere Meldungen, z.B. für teilweise Scharf „STATE 1 PART\_ARMED“. Details sind der Schnittstellenbeschreibung JA-121T\_English.pdf zu entnehmen.

## MAX3485:

Ermöglicht das Auslesen der RS485 Daten mittels der I/O Pins des ESP8266. Da die 5V Version MAX485 nicht verfügbar war, wurde der MAX3485 (SMD-Ausführung) eingesetzt und über einen 3V3 Pin des ESP8266 mit Spannung versorgt. Da das SO-Gehäuse des IC's für Tests nicht auf eine Breadboard steckbar ist wurde eine kleines Shield entwickelt. Details siehe EAGLE Board „MAX3485\_Adapter.brd“.

## ESP8266 und Sketch

Der Mikrokontroller empfängt die Daten ex JA-121T und sendet diese über die MQTT-Schnittstelle an den CCU-Jack:

- MQTT\_PUB\_JABLO\_RS485: Heartbeat alle 10 Minuten an die CCU3 durch Triggerung des Programmes „RS485 aktiv“
- MQTT\_PUB\_JABLO\_STATUS: Statusmeldungen Anlage und Sensoren
- MQTT\_PUB\_JABLO\_PROG: Trigger des HmIP Scriptes „Alarmanlagen\_Status\_ermitteln\_Vnn.hms“ zur Verarbeitung der Statusmeldungen.

Eine Steuerung der Alarmanlage über MQTT ist dzt. nicht geplant. Für Fehleranalysen werden die empfangenen Daten auf eine MicroSD-Karte gespeichert. Details sind dem Sketch „Jablotron-RS485-HomeMatic IP\_Vxx“ zu entnehmen.

Als Basis für die einzelnen Sketch-Teile wurden die folgenden Anleitungen verwendet:

- Fixe-IP: „ESP8266 NodeMCU Static/Fixed IP Address“ von Random Nerd Tutorials
- RS485: „How to interface Arduino, esp8266 or esp32 to RS-485“ von Renzo Mischanti
- MQTT: „ESP8266 NodeMCU MQTT“ von Random Nerd Tutorials
- Time-Server: „NTP DST for the ESP8266“ von Werner Rothschof
- SD-Card: „SD Card Module With ESP8266“ von Fernando Koyanagi

## Platinen:

Die Entwicklung der Platine wurde mit EAGLE ausgeführt. Details siehe EAGLE Schaltplan (.sch) und Board (.brd) „RS485\_Jablotron“ bzw. „MAX3485\_Adapter“.

## Gehäuse:

Maßgeschneidertes Gehäuse für die Hardware, entwickelt mit FreeCAD und gedruckt mit Prusa i3-mk3s. Details siehe Gehäuse\_RS485-ESP8266.FCStd.

## HomeMatic<sup>IP</sup>:

### CCU-Jack

Die Zusatzsoftware für die CCU3 empfängt und quittiert die Daten ex ESP8266. Installationsanleitung ist im HomeMatic Forum unter „CCU-Jack: REST-API/MQTT-Server/MQTT-CCU-Adapter“ zu finden.

Systemvariablen	Zweck	Wert
[Jablo] Devices	Liste der Jablotron Devices gemäß PRFSTATE Bitleiste sortiert	Namensliste
[Jablo] Devicestatus	„Bitleiste“ der aktiven/inaktiven Jablotron Devices	Zeichenkette
[Jablo] Kontakt offen	Eine oder mehrere Fenster/Türkontakte sind offen	wahr/falsch
[Jablo] RS485 aktiv	Die RS485 Schnittstelle zu Jablotron JA121-T ist aktiv	wahr/falsch
[Jablo] Status	Stellt den Status der Jablotron Alarmanlage dar	Zeichenkette

Programm/Script	Zweck
Alarmanlagen Status ermitteln	Bringt die RFSTATUS „Bitleiste“ in lesbare Form. Ermittelt den Status der Alarmanlage. Trigger über CCU-JACK MQTT program/set/42389
RS485 überwachen	Prüft, ob das die RS485 MQTT Schnittstelle noch läuft. Trigger über CCU-JACK MQTT program/set/39595
RS485 inaktiv	Fehlermeldung über Pushover, wenn Schnittstelle nicht mehr aktiv