# ESP32-Cam – QR-Code Scanner

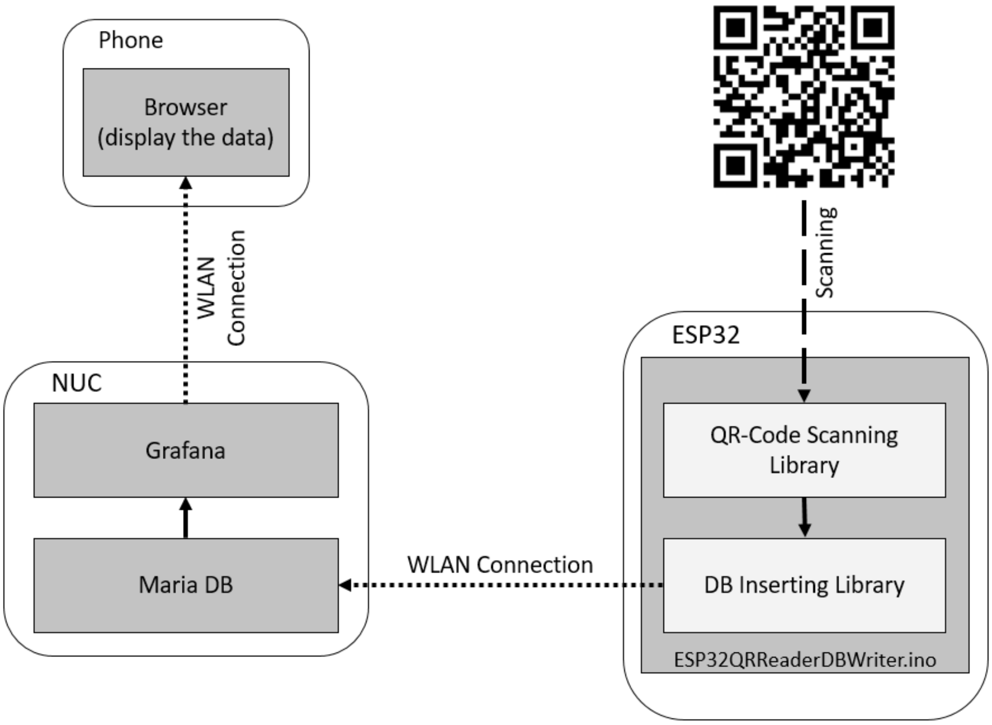
## Project aim:

The aim of the project is to set up a ESP32-Cam microcontroller in a way, that it can scan QR-Codes, which content then should be sent to a server and inserted to a database. The QR code scanner and the database should be set up according to the "plug and play" principle - this means on the one hand that the user does not need any special knowledge to operate the system, and on the other hand that scanning the data and reading and visualizing the data from the database should be very quick and easy.

## Hard- and Software components – overview:

As hardware components a ESP32-Cam and a NUC were used for scanning and storing the data. For displaying the data any commercially available “modern” phone can be used.
For the task of the  ESP32-Cam two libraries were used, which were combined in the program ESP32QRReaderDBWriter.ino. One library handles the task of scanning and reading/decoding the QR-Codes. This library is based on the Github-Code of Chung Yi Fu, which can be found at: https://github.com/alvarowolfx/ESP32QRCodeReader/tree/master.
The second library handles the task of connecting the ESP32 via W-LAN to a server on which a MariaDB is hosted and inserting the scanned data into the database. In our case the database is hosted by a Nuc. The code of the MariaDB-Insert is based on the Github-Code of  Dr. Charles A. Bell, which can be found at: https://github.com/ChuckBell/MySQL_Connector_Arduino.
In order to display the data, a cell phone – which connects to the Nuc vis WLAN – can be used. Therefore the browser of the phone connects to Graphana – which is hosted on the Nuc – which handles the task of displaying the data stored in the MariaDB graphically.

## Flow of the data:



# Setup of the System:

In order to start up the **ESP32**, network specific- and database specific data has to be set. I **credentials.h**: The network-name and the network-password has to be set, as well as the username and the password of the the database-user.
**ESP32QRReaderDBWriter.ino**: Regarding the database the port of the database, the name of the database and the specific table in the database, in which the QR-Code information should be inserted, has to be set. In addition the insert-String has to be adapted to the specific case (f.e. the column name).
Also the IP-adress of the server (Nuc) has to be set.
After all these parameters are set, the program can be uploaded to the ESP32, which is than ready for use.

In order to start up the **Nuc**, it only has to be supplied with power and the power-button has to be pressed. The Nuc is configured in a way that the W-LAN hotspot, the MariaDB, as well as Grafana start up automatically.
In Grafana a public dashboard was set up, in which displays the data of the database, when the following url is called:
http://10.42.0.1:3000/public-dashboards/2fe9455e3b5246728ce6f1e8f4837228

# File structure of the "ESP32 - QRCode Scanner – Projekt" folder:

In the "ESP32 - QRCode Scanner – Projekt"-folder include different projects:
**ESP32QRReaderDBWriter:** is the previously described project, where the QR-Code reader and the Database insert library are implemented and merged together.
**ESP32DatabaseInsert_SetUp:** this project only implements the library, which inserts data to the MariaDb.
**ESP32QRCodeReader_Page:** this project only implements the library, which scanns and decodes the QR-Code on the ESP32.
**ESP32_VideoStreamer:** this project allows the EPS32 to become a server which connects to a specific W-LAN and streams the video of the camera. The video stream can be retrieved by calling the URL of the server, which is hosted by the ESP32, in the browser.
**ESP32QRReaderDBWriterVideoStreamer:** This project combines the libraries of the QR-Code scanner, the Database-Inserter and the Videostreamer into one project. However, as it seems the computing power of the ESP32 is to low to realise all three functionalities at the same time. In order to be able to scan and read the QR-Codes the resolution of the video stream has to be set to such a low value, that the browser is not able to display the video stream any more. In addition the QR-Code scanner does also not work well in this case.
When increasing the resolution of the video stream, the browser is able to display the stream (which has a really(!) low frame rate), however the QR-Code reader is not able to decode the QR-Codes any more.

## NUC WLAN-configuration:

In order for the W-LAN hotspot to start automatically and to enable the connection from all (tested) devices, a configuration in the related file has to be changed.
The networkconfigurations are stored in the file: *etc/NetworkManager/system-connections/Hotspot.nmconnection*
In order to enable the autostart of the W-LAN hotspot, the parameter autoconnect has to be set on true.
In order for all (tested) phones to be able to connect to the NUC, the proto parameter has to be set on wpa. However, in this case the ESP32 microcontrollers are not able to connect to the W-LAN. When changing this setting to rsn than the ESP32 microcontroller can connect to the W-LAN, however not alle phones can connect to it anymore.
The current solution is to disable the W-LAN security, by deleting the [wifi-security] block from the configuration file. In this case the mobile phones, as well as the ESP32 can connect to the W-LAN of the NUC.
In order to be able to connect the ESP32 to the unsecured W-LAN the code line in the .ino-File “WiFi.begin(ssid, pass);” has to be changed to “WiFi.begin(ssid);”.