



HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

Praktikum im Fach Internet of Things (IoT) / Industrie 4.0

Versuch 1: Haustür Überwachung im Smart Home

Autoren: Marco Schaarschmidt (m.schaarschmidt@hs-osnabrueck.de)
Nicolas Lampe (n.lampe@hs-osnabrueck.de)
Version: 2.1

Wichtige Hinweise:

- Zu dieser Aufgabenstellung gehört ein Dokument mit Basisinformation.
- Bei Fragen oder Rückmeldungen zum Praktikumsversuch stellen Sie diese bitte primär zu den Praktikumszeiten (dienstags 14:30 bis 16:00). Bei dringenden Fällen schreiben Sie eine E-Mail an oder Nicolas Lampe (n.lampe@hs-osnabrueck.de) oder Sebastian Böhm (sebastian.boehm@hs-osnabrueck.de).

1 Einleitung

Neben der Anwendung des Internet of Things (IoT) beispielsweise in den Bereichen Agrartechnik, der Mobilität oder der Logistik ist ein stetig wachsender Anwendungsbereich des IoT die Verwendung für Smart Home Applikationen. Durch eine zunehmende Vernetzung von Haushaltsgeräten und allgemein der Haustechnik soll der Alltag einfacher gestaltet werden. Zu den Vorteilen von Smart-Home Anwendungen zählen ein verbesserter Komfort, erweitertes Entertainment, erhöhte Sicherheit und ein verringerter Energieverbrauch. Dabei können einzelne Produkte beispielsweise über WLAN, Bluetooth, ZigBee oder weitere Protokolle kommunizieren. Aufgrund deren Vernetzung gelten diese als IoT Anwendungen. Eine weitere Komponente von Smart-Home ist die Automation mittels Sensoren und Aktoren. Beim Smart-Home gibt es häufig ein zentrales Bedienelement (Gateway), welches ein Touchdisplay an der Wand, ein Sprachassistent oder eine Smartphone-App sein kann.

Ziel dieses Versuches ist es, mit der *M5Stack Timer Camera* und dem *M5Stack* ein System zur Haustürüberwachung zu entwickeln. Im ersten Teilversuch soll vom Smartphone aus der Videostream der Kamera abgerufen und angezeigt werden. Im zweiten Teilversuch soll der Videostream auf dem Display des M5Stacks angezeigt werden. Für diesen Versuch benötigen Sie folgende Hardware des IoT/I4.0-Box:

- M5Stack
- M5Stack Timer Camera

Generell sollten Sie vor Beginn dieses Versuches die Basisinformationen gelesen und ebenfalls verstanden haben, damit Sie die Grundlagen zur Bearbeitung des Versuches besitzen und der Einstieg in die Entwicklung mit dem M5Stack gelingt.

2 Versuchsdurchführung

Die Versuchsdurchführung setzt sich zusammen aus zwei Teilaufgaben. In der ersten Teilaufgabe soll die *M5Stack Timer Camera* mit dem Smartphone verbunden werden. Anschließend soll in der zweiten Teilaufgabe die Verbindung des *M5Stacks* mit der *M5Stack Timer Camera* hergestellt werden, wobei das Video auf dem Display des *M5Stacks* über WiFi gestreamt werden soll. Als Versuchsprotokoll sind die nachfolgenden Fragen in schriftlicher Form zu beantworten und mit dem entwickelten Programmcode in OSCA vor dem Testat zu hinterlegen:

- Erklären Sie kurz die Funktionsweise eines Access Points. Warum ist es sinnvoll bei dem Anzeigen des Kamerabilds auf dem M5Stack die Access Point Funktion zu verwenden?
- Welche Funktion hat der http-Header?
- Begründen Sie, warum eine Vorverarbeitung der empfangenden Bilder im Rahmen dieses Praktikums nicht benötigt wird. Was müssten Sie an Ihrer Lösung anpassen, wenn Sie z.B. ein „FullHD“-Bild empfangen?
- Begründen Sie, warum gerade Buffer für größere Datenmengen oder mit unbekannter Größe auf dem Heap und nicht auf dem Stack angelegt werden sollen?
- Auf welche Schwierigkeiten sind Sie bei der Versuchsdurchführung gestoßen?

2.1 Inbetriebnahme der M5Stack Timer Camera

Um die *M5Stack Timer Camera* nutzen zu können, muss zunächst die zur Verfügung gestellte Software auf den ESP32 des Kameramoduls gespielt werden. Befolgen Sie dabei die nachfolgende Anleitung zur Inbetriebnahme. Machen Sie sich mit der grundlegenden Funktionalität des Quellcodes vertraut. Ein tiefes Verständnis ist jedoch nicht vonnöten und nicht Bestandteil des Testates zu diesem Versuch. Eigenständige Anpassungen an den Einstellungen der Kamera sind für die Durchführung des Versuches nicht notwendig.

Laden Sie sich in einem ersten Schritt das zur Verfügung gestellte Archiv *CameraWebServer* herunter und entpacken Sie diesen auf Ihrem PC bzw. auf dem Pool-PC. Im nächsten Schritt ist die Datei *CameraWebServer.ino* in der Arduino IDE zu öffnen.

Vorab können Sie zwischen zwei Optionen für den Betrieb des Kameramoduls wählen. Wird der nachfolgende Parameter

```
#define RUN_IN_AP_MODE true/false
```

auf den Wert *true* gesetzt, läuft das Kameramodul im Access Point Modus und spannt ein eigenes WLAN-Netz auf, über das Sie sich mit Ihrem Smartphone oder mit dem M5Stack verbinden und auf den Videostream der Kamera zugreifen können. Ist der Parameter auf den Wert *false*

gesetzt, verbindet sich das Kameramodul mit dem angegebenen WLAN. Unabhängig von dem gewählten Betriebsmodus müssen die Variablen

```
const char* ssid = "M5Stack-Timer-Camera";  
const char* password = "Password";
```

von Ihnen entsprechend gesetzt werden. Mit der *ssid* und dem *password* spezifizieren Sie die Eigenschaften des WLAN-Netzes. Im Falle des Access Point Modus sind dies die Eigenschaften des neu aufgespannten Netzwerkes. Soll sich das Kameramodul mit einem bestehenden WLAN-Netzwerk verbunden werden, so sind als *ssid* der Name und als *password* entsprechend das Passwort des lokalen WLAN-Netzwerks anzugeben. Bitte bedenken Sie, dass eine Verbindung mit eduroam Netzwerk nicht möglich ist.

Im nächsten Schritt wird die Software über die Arduino IDE auf die *M5Stack Timer Camera* überspielt. Dazu schließen Sie zunächst das Kameramodul über das USB-C Kabel an Ihren PC an. Bei korrekter Installation der Toolchain (siehe Dokument mit Basisinformationen), können Sie die vordefinierte Boardbeschreibung über *Werkzeuge → Board → ESP32 Arduino → M5Stack-Timer-CAM* auswählen. Im Anschluss wählen Sie über *Werkzeuge → Port* den korrekten COM-Port aus. Beim Hochladen der Kamera-Software wird auf dem seriellen Monitor (*Werkzeuge → Serieller Monitor*) der Status ausgegeben, sowie die IP-Adresse des Kameramoduls, die für die kommenden Schritte benötigt wird.

2.2 Smartphone mit der M5Stack Timer Camera verbinden

Sofern die *M5Stack Timer Camera* mit Strom versorgt wird (möglich über einen Anschluss an den PC, eine Power Bank, ...) und die Software aufgespielt worden ist, kann der Kamerastream entweder über den konfigurierten Access Point oder Ihr lokales WLAN und der auf dem *Seriellen Monitor* angezeigten IP-Adresse erreicht werden. Mit Eingabe der IP-Adresse im Browser kann der Video-Stream angezeigt werden. Das von der Kamera aufgenommene Bild kann durch Eingabe der IP-Adresse unter Hinzunahme der Endung „<IP-Adresse>:80/capture“ aufgerufen werden. Für den zweiten Teil des Versuches soll die Verbindung über den Browser wieder beendet werden, da ein Zugriff auf den Stream durch mehrere Geräte nicht unterstützt wird.

2.3 M5Stack mit der M5Stack Timer Camera verbinden

Für den zweiten Teilversuch soll ein erstes Programm erstellt und auf den M5Stack aufgespielt werden. Legen Sie dazu zuerst einen neuen Sketch in der Arduino IDE an und starten Sie dann mit der Entwicklung. Das Initialisieren der WiFi-Verbindung soll in der Funktion *setup()* erfolgen. In der *loop()* soll das Programm auf die erfolgreiche WiFi-Verbindung warten. Anschließend soll eine http-Verbindung aufgebaut werden. Die http-Verbindung muss ebenfalls nach dem Status abgefragt werden. Danach muss die Antwort verarbeitet werden, sodass sie auf dem Display ausgegeben werden kann.

Hinweise:

- Für eine erhöhte Anzahl an Bildern pro Sekunde auf dem M5Stack bietet es sich die Funktion Access Point zu verwenden.
- Für das Programm sind die Libraries „M5Stack.h“, „WiFi.h“ und „HTTPClient.h“ einzubinden.
- Nutzen Sie (in chronologischer Reihenfolge) vom *HTTPClient* die Funktionen *begin* zur Angabe der URL und zum Starten der Anfrage, *GET()* zum Abfragen der URL, *getSize()* um die Größe der auszulesenden Daten zu ermitteln sowie *getStreamPtr* um einen Pointer auf den Stream zu erhalten. Mit dem Stream können Sie dann über *readBytes* den Inhalt auslesen.
- Beim Anlegen von Buffern nutzen Sie den *malloc*, für das Freigeben den *free* Befehl
- Um das empfangene Bild auf dem Display anzuzeigen soll die Funktion *drawJpg()* „M5Stack“-Bibliothek genutzt werden