WPF-Projekt:

Wetterstation

Von

Manuel Yates (AW)

Felix Becker (SI)

Michael Schkalei (AW)

Peter Rietz (AW)

Lukas Sänger (AW)

Luca Grethen (SI)

Zuletzt geändert am:

15.07.2022

# Vorwort

Das Projekt wurde in eigenständiger Arbeit geplant, vorbereitet und durchgeführt.

# Projektbeschreibung

Anforderung war es eine vollständige Umgebung einer Wetterstation zu erstellen.  
Diese sollte folgende Funktionen beinhalten:  
- Aufnahme der Sensordaten von „Octopus“-Boards über das MQTT Protokoll  
- Speicherung der aufgenommenen Daten in einer Datenbank  
- Visualisierung über eine Internetseite

Es gab keine Einschränkungen in Hinblick auf verwendete Architektur, Entwicklungssprachen und verwendete Programme.

Die gesamte Projektarbeit fand unter Eigenarbeit ohne konkrete Aufsicht eines Lehrers statt.  
Zusätzlich zur Umsetzung des Projektes wurden einige Unterrichtsstunden zur Vorstellung von Programmiersprachen, Architekturen sowie auch Datenbank-Grundlagen verwendet.

# Projektumsetzung

## Aufgabenverteilung

Manuel Yates: Projektleitung, Leiter der Softwareentwicklung

Felix Becker:

Michael Schkalei:

Peter Rietz: Software-Entwicklung, Dokumentation

Lukas Sänger:

Luca Grethen:

## Projektumgebung (Hardware)

Zur Aufnahme von Sensordaten wurden die sogenannten „Octopus-Board“ der BBS verwendet.  
Hierbei handelt es sich um Microcontroller Boards, welche Standardmäßig über einen Temperatur, Luftdruck und Luftfeuchtigkeitssensor verfügen. Zusätzlich können diese über WLAN über das MQTT-Protokoll kommunizieren und Daten austauschen.

Zusätzlich wurde in der finalen Version ein Raspberry Pi 4 mit 2GB Arbeitspeicher zum Speichern der Daten und zum Hosten des Webservers verwendet.  
  
Abseits davon wurden lediglich USB Kabel zur Stromversorgung und Netzwerkkabel für die Kommunikation verwendet.

## Projektumgebung (Software)

Für die Entwicklung der Software auf dem Octopus Board wurde die Software „Arduino-Studio“ in Kombination mit der Erweiterung „Ardu-Block“ verwendet. Hierbei handelt es sich um eine Erweiterung die eine „Low-Code“ Programmierung mit Hilfe von Blöcken ermöglicht. So können schnell und übersichtlich Programme erstellt, kompiliert und auf die Octopus Boards geladen werden. Zusätzlich ermöglicht es einen leichten Einstieg in die Programmlogik, ohne vom Anwender die benötigte Programmiersprache zu verlangen.

Als Betriebssystem auf dem Raspberry Pi kommt „Raspberry Pi OS Lite“ in der 64 Bit Variante zum Einsatz. Dieses Betriebssystem ist eine modifizierte Version der Linux-Distribution „Debian“, welche Weltweit einen Großteil der auf Server verwendeten Betriebssystemen ausmacht. Raspberry Pi OS Lite bietet sich in diesem Kontext hervorragend an, da auf eine Desktop Oberfläche vollkommen verzichtet wird und so Ressourcen eingespart werden. Der Zugriff auf das Betriebssystem findet über anstelle über das Netzwerkprotokoll SSH statt. Dieses bietet einen gesicherten Zugriff auf den Raspberry Pi und stellt die Systemkonsole (Terminal) zur Verfügung.  
Über das Terminal können alle wichtigen Schritte für die Einrichtung der benötigten Programme vollzogen werden.

Als MQTT Broker wurde die Software „Mosquitto“ von Eclipse verwendet. Hierbei handelt es sich um einen der verbreittesten MQTT Broker. Die Funktionalität des Brokers wird im Verlaufe dieser Dokumentation noch weiter behandelt.

Da eine direkte Kommunikation zwischen MQTT Broker und Datenbank nicht möglich ist, wurde das Programm „NodeRed“ verwendet. Hierbei handelt es sich um ein „Workflow“-Tool welches häufig im IoT Bereich Verwendung findet. Mit diesem Tool können vordefinierte Abläufe realisiert und gesteuert werden. Das Programm verwendet wie auch ArduBlock einen „Low-Code“ Ansatz. Hier können über kleine Blöcke und Pfeile gezielte Abläufe optisch strukturiert realisiert werden.

Als Datenbank wurde eine MariaDB mit MySQL verwendet. Dieses relationale Datenbanksystem bietet sich an, da es problemlos unter Linux funktioniert und einen guten Kompromiss aus Geschwindigkeit und Benutzerfreundlichkeit gibt. Die verwendete Datenbanksprache ist SQL.

Als Verwaltungssoftware wurden PhpMyAdmin und MySQLWorkbench verwendet. PhpMyAdmin ist ein Standard Dienst der XAMPP Programmfamilie. Über die Weboberfläche können schnell und übersichtlich Zugriffe auf die Datenbank stattfinden. MySQLWorkbench ist ein Desktop-Programm zur Verwaltung von MySQL Datenbanken. Im Vergleich zu PHPMyAdmin ist hier die Benutzer- und Einsteigerfreundlichkeit geringer, dafür aber die Zugriffs- und Verwaltungsmöglichkeiten höher.

Als Programmiersprache wurde die objektorientierte Programmiersprache C# verwendet. C# wird von Microsoft entwickelt und befindet sich aktuell in der Version 10. Mit C# können neben Desktop (WPF) und Mobile-Applikationen (Xamarin) auch Webdienste (ASP, Blazor) und Multiplattform-Programme (.NET MAUI) realisiert werden. Verwendet wird Sie ebenfalls in Bereichen wie Machine-Learning (ML.NET) und Spieleentwicklung (Unity-Engine) verwendet. Sie gehört damit zu den zukunftssichersten und populärsten Programmiersprachen und ist laut „Tiobe-Index“ Platz 5 der am meisten verwendeten Programmiersprachen.

Im Projekt wurde das .NET Framework in der Version 6 verwendet. Dies ist die aktuellste Version und beinhaltet viele Hilfreiche Bibliotheken und Funktionen.

Als sogenannte IDE (Integrated Development Environment) wird Visual Studio Community 2022 verwendet. Diese IDE ist speziell auf C# zugeschnitten und bietet neben Debugging einen großen Pool an nützlichen Funktionen und Tools.

Da die Wetterstation über eine Oberfläche im Internet verfügen sollte, wurde eine „Blazor ServerSide App“ verwendet. Die Komponenten-Frameworks „MudBlazor“ sowie „RadzenBlazor“ wurden zur Darstellung der Oberfläche verwendet. Für die Anbindung an die Datenbank kam das Entity Framework als Objekt-Relationaler-Mapper zum Einsatz.

Als Versionsverwaltung kam Git über Github zum Einsatz. Git ist ein weit verbreitetes Open-Source-Tool zur verteilten Versionskontrolle von Software. Es wird sowohl in Unternehmen als auch von privaten Entwicklern weltweit genutzt und funktioniert zusammen mit verschiedensten Plattformen und IDEs. Über dieses Tool ist ein einfaches und übersichtliches Verwalten von Programmcode und Änderungen an diesem möglich. GitHub ist ein gewinnorientiertes Unternehmen, das einen Cloud-basierten Git Repository Hosting Service anbietet. Seit 2018 gehört das Unternehmen zu Microsoft.