Beschreibung der Softwareentwicklung

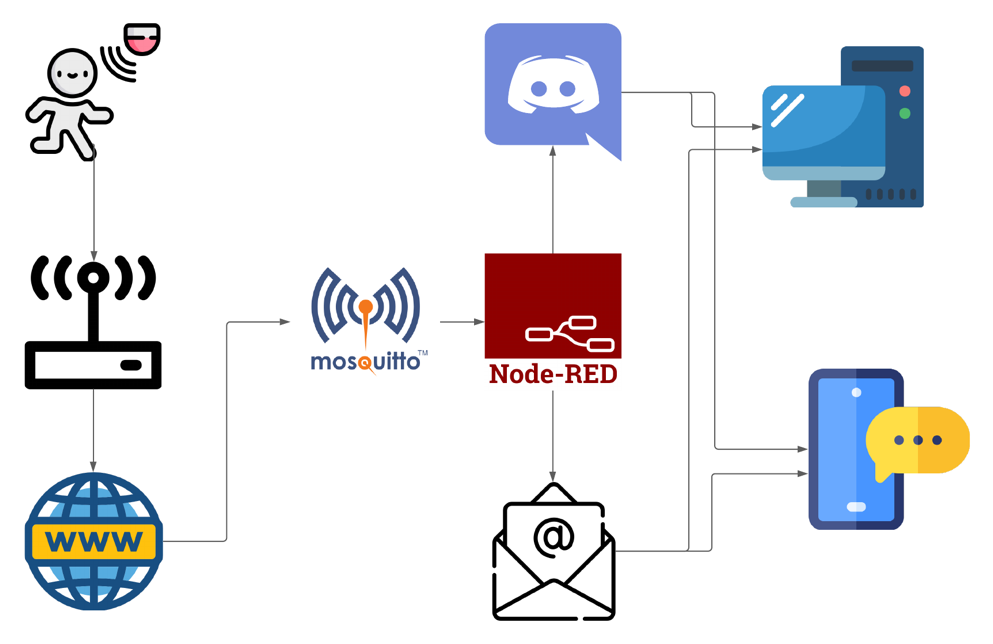


Abbildung 1/ Systemübersicht

Das Sicherheitssystem stellt bei Inbetriebnahme eine Verbindung zu einem spezifizierten Netzwerk her. Dies ist notwendig, um eine Kommunikation zwischen den Systemen zu gewährleisten. Erkennt das Sicherheitssystem eine Bewegung, wird eine entsprechende Meldung via MQTT über das Internet übermittelt. Die Daten werden von einem Broker (Mosquitto) empfangen und an einen Node-RED Server weitergeleitet.

Die empfangenen Daten werden mittels Node-RED als Email-, sowie Discord-Nachricht verarbeitet und weitergeleitet. Die verarbeitete Meldung erscheint schließlich auf den jeweiligen Endgeräten.

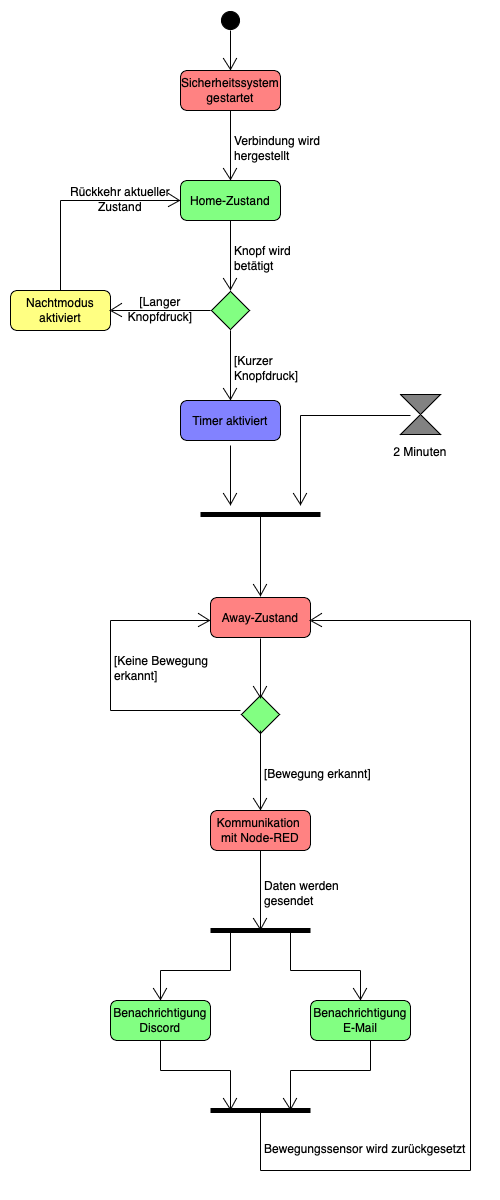


Abbildung2/ Zustandsdiagramm

Bei Start des Sicherheitssystems wird beginnend das Setup des Programms ausgeführt. In dieser Phase werden Verbindungen zum Netzwerk, zum MQTT-Broker sowie zum Node-RED Client hergestellt und aufrechterhalten.

Nachdem alle Verbindungen erfolgreich hergestellt werden konnten, beginnt die „Loop-Methode“ des Programms. In diesem Teil des Programms wird die Logik des Buttons und des Sicherheitssystems umgesetzt.

Die von Spivey entwickelte Button-Logik nutzt Zustände und Timer, um zwischen kurzen und langen Tastendrücken zu unterscheiden. Dies ist für die spezifische Funktionalität des Sicherheitssystems notwendig. Erkennt das System einen kurzen Knopfdruck, wechselt das System abhängig vom vorherigen Zustand in den „Home“- oder den „Away“-Zustand. Registriert das System einen langen Knopfdruck, wird der Nachtmodus des Sicherheitssystems aktiviert. Die LED-Leuchten werden deaktiviert und der aktuelle Zustand bleibt bestehen.

Das eingeschaltete System befindet sich im Ausgangszustand „Home“. In diesem Zustand wird lediglich die grüne LED-Leuchte aktiviert. Es werden in diesem Zustand keine Bewegungsdaten erfasst und ausgewertet.

Nachdem das System in den „Away“ -Zustand gelangt, erlischt die grüne LED-Leuchte und die rote LED wird aktiviert. Mithilfe eines im Setup definierten Zählers „counter“, kann zwischen Zuständen vor und nach der Timerlogik separiert werden. Dabei wird bei erstmaligem Eintritt in den „Away“-Zustand durch den vordefinierten Zähler das Sicherheitszeitintervall von zwei Minuten gestartet. Nachdem der Timer abgelaufen ist, gelangt man durch das Inkrementieren des Zählers in den Folgezustand, welcher die Bewegungssensorik aktiviert.

Erkennt das Sicherheitssystem eine Bewegung, wird eine entsprechende Meldung an den MQTT-Broker übermittelt. Nach einer kurzen Verzögerung wird der Zähler wieder zurückgesetzt und das Programm gelangt wieder in den Zustand, welcher für die Bewegungserkennung verantwortlich ist. So kann eine Überwachung ohne erneute Einhaltung des Sicherheitszeitintervalls gewährleistet werden.

Die Datenübertragung wird über einen mit Node-RED eingerichteten Server abgewickelt. Innerhalb des Programms wird ein vordefiniertes Topic („MFHLSensor“) genutzt, um eine Kommunikation über den MQTT Broker zu ermöglichen. Bei eintretender Bewegungserkennung wird mit der „publishString“ Methode eine Kommunikation zwischen den Schnittstellen eingeleitet. Nachdem die Daten an Node-RED übermittelt worden sind, muss eine Weiterleitung an die entsprechenden Endgeräte stattfinden.

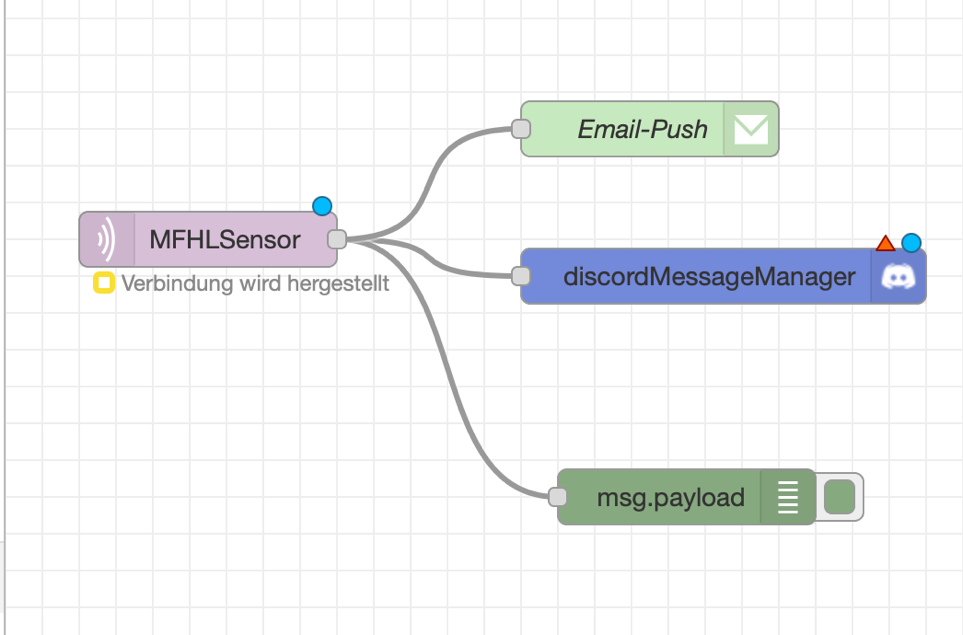


Abbildung 3/ Node-RED Konfiguration

Über das Netzwerk-Node „mqtt-in“ kann durch das Festlegen eines vorspezifizierten Servers, sowie Topics ein Dateninput verarbeitet werden. Dabei werden die Nodes „discord“ und „email“ zur Weiterleitung einer Push-Benachrichtigung genutzt.

Um eine Push-Benachrichtung an eine E-Mail Adresse weiterzuleiten, muss das „email“ Node lediglich mit Server, Port und den ID-Daten des Nutzers synchronisiert werden.

Um eine Push-Benachrichtigung via Discord-Bot durchzuführen, muss eine Applikation, sowie ein Discord-Bot initialisiert werden. Über das Developer-Portal von Discord kann eine entsprechende Applikation erstellt werden, welche das Verwenden eines Discord-Bots ermöglicht.

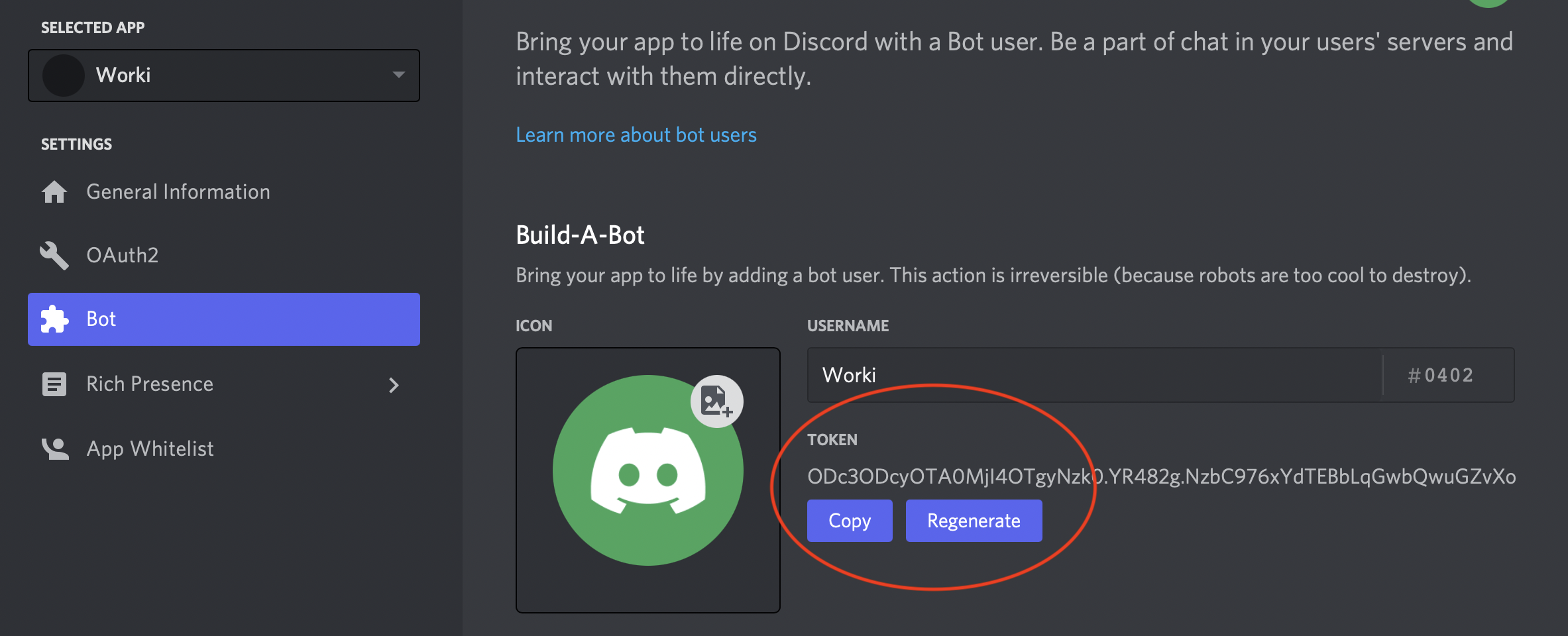


Abbildung 4/ Discord Developer Portal

Der erstellte Discord-Bot wird mit einem gegebenen Token über Node-RED initialisiert. So kann eine Kommunikation zwischen den Systemen stattfinden und eine Meldung an die Endgeräte weitergeleitet werden.