Folie 1

Herzlich willkommen zu unserer Präsentation. Wir stellen Euch heute unser IoT-Projekt vor.

Folie 2

Zunächst zum Ablauf der Präsentation.

<klick>

Wir erläutern zunächst die Problemstellung und stellen Euch unsere Lösungsidee vor.

<klick>

Anschliessend werden wir genauer auf unser Projekt eingehen.

Dazu zählt beispielsweise das Vorstellen der benutzten Sensoren und Technologien. Ausserdem werden wir wichtige Code-Überlegungen, sowie unser Dashboard vorstellen. Anschliessend folgt unser Vorgehen in Bezug auf die Elektronik.

<klick>

Abschliessend werden wir noch einige weiterführende Ideen vorstellen, die leider für dieses Projekt Out-of-Scope waren.

Folie 3

Gas kann ein unsichtbarer Feind für uns Menschen sein.

Immer wieder wird in Zeitungen von Unfällen berichtet, in denen Menschen beispielsweise durch zu tiefe Sauerstoff- oder zu hohe Kohlenmonoxid-Werte zu Tode gekommen sind.

<klick>

Insbesondere in luftdichten Räumen, aber beispielsweise auch in Höhlen oder im Tagbau besteht die Gefahr von Gas-Vergiftungen oder einer zu tiefen Sauerstoffkonzentration. Leider werden diese oft erst bemerkt, wenn es bereits zu spät ist und die Menschen das Bewusstsein verloren haben.

Folie 4

Daher stellen wir heute gerne eine mögliche Lösung vor:

<klick>

Unser kleines Fahrzeug kann genutzt werden, um die Umgebung zu erkunden

<klick>

und dabei kontinuierlich die Gas-Konzentration in der näheren Umgebung zu messen.

<klick>

Die Gas-Konzentration wird dann in einer Datenbank gespeichert, so dass man diese auch analysieren könnte, um beispielsweise zu sehen, bei welchen Arbeiten welche Gefahren entstehen könnten.

<klick>

Die Gas-Konzentration wird aber auch in einem Dashboard angezeigt, so dass sie nahezu real-time überwacht werden kann

<klick>

und bei einer Gefahr, wird eine Whatsapp-Nachricht an die Mitarbeitenden geschickt.

<klick>

Das Fahrzeug selber verfügt ebenfalls über ein Alarmsystem, so dass die jeweiligen LED rot leuchten, wenn der betreffende Sensor den definierten Schwellenwert über- beziehungsweise unterschreitet. Ausserdem erzeugt ein Buzzer auch ein akustisches Signal.

Folie 5

Um unsere Lösungen umzusetzen haben wir folgende Sensoren und Technologien genutzt.

<klick>

Zum einen läuft die Software-Lösung auf einem Raspberry Pi.

<klick>

Dieses haben wir an den PiCar von Sunfounder angeschlossen, mit dem wir die Umgebung erkunden können.

<klick>

Als Gas-Sensoren nutzen wir vorkalibrierte Sensoren von DFRobot.

<klick>

Wir haben uns für die Demonstration für Ammoniak-, Kohlenmonoxid- und Sauerstoff-Messung entschieden. DFRobot hat aber noch weitere Sensoren und man kann diese recht einfach ersetzen. Auch in der Software müssten nur wenige Anpassungen vorgenommen werden.

<klick>

Für das Dashboard verwenden wir NodeRed, da wir diese Technologie im Rahmen des Moduls kennengelernt haben und es uns überzeugt hat.

<klick>

Für die Orientierung im Raum haben wir noch einen 2D-Lidar von LDRobot in die Lösung eingebaut, aber leider war die sinnvolle Nutzung von diesem dann Out-of-Scope.

Folie 6

Prototypisieren mit Notebooks, nachher umschrieben zur Python-Library.

Service auf Raspberry Pi, der die Software mit Sensor-Messung & PiCar-Steuerung automatisch laufen lässt.

Folie 7

Nun erläutern wir gerne noch die wichtigsten Überlegungen zum Software-Code.

<klick>

Um die Sensoren zu nutzen, braucht es die Library von DFRobot. Leider haben wir feststellen müssen, dass es extrem viele Fehler gibt, die das Programm dann abbrechen lassen. Da es sich bei den Sensoren um Hardware handelt, ist es möglich, dass einige Messungen nicht immer funktionieren. Diese sollten dann einfach übersprungen werden und nicht das ganze Programm abbrechen. Daher haben wir uns entschieden, die Library zu klonen und für unser Projekt anzupassen. So konnten wir nur den Teil aus der Library rausnehmen, den wir gebraucht haben.

<klick>

Für den PiCar haben wir die Library von SunFounder verwendet und dort beim picarx-controller einige Funktionen angepasst, so dass der PiCar beispielsweise vor einer Wand langsamer wird, damit man nicht mit vollem Tempo hineinfährt.

Folie 8

Für die Datenverarbeitung haben wir einen Measurement-Loop erstellt, der alle 0.1 Sekunden die Sensoren ansteuert und die Gas-Konzetration misst.

Die Werte werden nach diesem Measurement-Intervall überprüft und mit den festgelegten Alert-Level verglichen.

Die gelesenen Daten werden in einer Liste gespeichert und alle 0.5 Sekunden aggregiert. Dabei wird der Timestamp übergeben, sowie der Minimal- und Maximal- und der Durchschnittswert berechnet.

Die aggregierten Daten werden dann in die MongoDB geladen.

Dies ist nötig, damit die unterschiedlichen States in unserer Software funktionieren.

Für die Zukunft müsste man sich allenfalls noch ein Handling überlegen, wie man mit älteren Daten umgeht. Allenfalls könnte man Tages-Durchschnitte berechnen und in eine separate Collection speichern und alte Raw-Daten dann löschen, damit die Datenbank nicht überflutet wird.

Folie 9

Ein wichtiger Teil des Codes ist auch der AlertManager. Dieser verwaltet den Setup, wo die Adressierung der Elektronik-Komponenten festgelegt wird. Ausserdem werden im AlertManager die Schwellenwerte definiert und überprüft, ob sie über- beziehungsweise unterschritten werden.

<klick>

Für den Alarm haben wir 3 unterschiedliche Modi definiert.

Der erste Modus ist der normale Modus. Dabei leuchtet die grüne LED und die roten LEDs und der Buzzer sind ausgeschaltet. So weiss man, dass das Gerät an ist und die Software läuft.

Bei einem Alarm, wird die grüne LED ausgeschaltet und die betreffenden roten LEDs blinken. Auch der Buzzer ist dann an. Bei einem Alarm hat man die Möglichkeit, den Switch zu drücken, so dass der Zustand in den Acknowledge-Modus wechselt. In diesem ist die grüne LED immer noch aus und die betreffenden roten LEDs leuchten statt zu blinken. Ausserdem ist der Buzzer wieder ausgeschaltet. Alle 0.4 Sekunden wird überprüft, ob sich die Gas-Konzentration immer noch im kritischen Bereich befindet. Falls nicht, wird wieder auf den Normal-Modus zurückgesetzt.

Folie 10

Nun kommen wir zur Elektronik.

Ursprünglich haben wir die Prototypisierung mit einem Breadboard gemacht.

<klick>

In der ersten Iteration haben wir nur einen Sensor angeschlossen und auch dementsprechend nur 1 rotes und 1 grünes LED.

<klick>

In der 2. Iteration haben wir dann alle drei Sensoren genutzt und überprüft, ob die LEDs einzeln leuchten aber ob es auch möglich ist, dass mehrere gleichzeitig Alarm anzeigen.

<klick>

In der 3. und letzten Iteration haben wir dann alle Sensoren und auch den Buzzer und den Switch angeschlossen.

Doch obwohl wir versucht haben, die Kabel so kurz wie möglich zu halten, waren wir mit dieser klobigen Lösung unzufrieden. Daher haben wir uns entschieden, unsere eigene Platine zu designen.

Folie 11

Die Schwierigkeit war, dass wir eigentlich nur die 3 Gas-Sensoren, nicht aber den Lidar und den PiCar auf dem Board testen konnten. Daher wussten wir, dass wir nicht alle Pins nutzen dürfen, da wir die Erweiterung mit dem PiCar und dem Lidar unbedingt offenbehalten wollten. Ausserdem kann man theoretisch 4 Sensoren von DFRobot parallel nutzen, daher wollten wir auch einen 4. Sensor im Elektronik-Design einbinden.

<klick>

Die Lösung für die Schwierigkeiten war, dass wir die Dokumentation vom PiCar und vom Lidar nutzen mussten, um zu erfahren, welche Pins gebraucht werden. Glücklicherweise gab es beim PiCar und beim Lidar keine Überschneidungen.

<klick>

Als wir die Pins auf dem Design entsprechend angeschrieben und blockiert hatten, haben wir für die restlichen 4 Sensoren die Pins verteilt

Folie 12

Dies hat dann so ausgesehen.

Das Design wurde mit KiCAD erstellt. Im Schemata-Editor kann man alle Komponenten zeichnen und dem Board zuordnen. Verbindungen werden dann erst im nächsten Schritt gemacht.

Folie 13

Nachdem das Design fertiggestellt war, konnte man im PCB-Editor das eigentliche Design erstellen. Wir haben beschlossen, dass wir uns auf 2 Layer beschränken, da wir grundsätzlich einen einfachen Use-Case haben und 2 Layer deutlich günstiger sind.

<klick>

Die Herausforderung war, dass wir überprüfen mussten, dass die gewählten Footprints stimmen, damit wir die richtige Grösse der Komponenten haben. Ausserdem mussten wir genau messen, um die Schraub-Löcher und die Pins am richtigen Ort zu haben.

Da wir das Board vom PiCar auf unser Design raufstecken wollten, mussten wir auch sicherstellen, dass höhere Komponenten, wie beispielsweise der Transistor oder der Buzzer nicht mit der Elektronik des PiCar-Boards kollidieren.

Folie 14

Die Lösung sah am Schluss so aus.

Folie 15

Als das Design dann fertig war, haben wir uns informiert, wo man es bestellen kann.

<klick>

Anschliessend haben wir 10 PCBs bei PCB-Way in China bestellt.

<klick>

Die Elektronik-Komponenten haben wir bei DigiKey bestellt.

<klick>

Und anschliessend auf die Boards gelötet.

Folie 16

Beim Löten kamen wir an folgende Herausforderungen:

<klick>

Beim ersten Lötversuch stellten wir fest, dass die Komponenten durchaus auch schräg gelötet werden können, wie man auf dem Bild sieht.

<klick>

Als Lösung dafür haben wir Tape genutzt, um die Komponenten zu befestigen.

<klick>

Die zweite Herausforderung war, dass wir hauptsächlich mit bleihaltigem Löt-Zinn gearbeitet haben, da dieser insbesondere für schwierigere Lötstellen, wie die Pins oder die Löt-Stellen des Transistors geeigneter war als bleifreier Löt-Zinn. Da er aber giftig ist, mussten wir uns eine Lösung überlegen.

<klick>

Daher wurde auf meinem Balkon eine gut durchlüftete Löt-Station eingerichtet und als zusätzlicher Schutz noch eine Brille und eine Atemschutz-Maske verwendet.

Folie 17

Folie 18

Folie 19

Nun kommen wir noch zu den Ideen, die leider Out-of-Scope waren.

<klick>

Zum einen hätte man den PiCar selbstständig den Raum erkunden lassen können. Dazu hätte man den Lidar-Sensor für die Raum-Erfassung nutzen können, damit der PiCar gezielter den Raum erkundet.

<klick>

Ausserdem wäre es cool gewesen, wenn der PiCar bei einer auffälligen Gas-Konzentration, versucht hätte, herauszufinden, woher das Gas kommt und sich dann immer näher zum Gas-Leck bewegt hätte.

<klick>

Ausserdem hält der Akku des PiCars nicht sehr lange, daher wäre es sicherlich noch sinnvoll, sich in Zukunft Gedanken zum Energie-Management des Fahrzeugs zu machen. Allenfalls könnte man noch weitere Batterien anschliessen oder allenfalls einen Arduino statt eines Raspberry Pis verwenden.

<klick>

Das Daten-Management in der DB haben wir ja bereits angesprochen. Es würde auch dort noch weitere Optimierungen geben.

Folie 20

Damit sind wir am Ende unserer Präsentation und würden nun gerne eine Live-Demo unserer IoT-Lösung demonstrieren.

Wir haben dafür ein Video aufgenommen, dass den gesamten Scope zeigt.

Für die Vorführung im Klassenzimmer haben wir uns überlegt, dass wir einfach kurz die Fahrweise, sowie den O2-Sensor und unser Dashboard demonstrieren.

Wir hätten aber auch Räucherstäbchen und Ammoniak dabei, würden aber empfehlen, dass wir dieses nicht im Unterrichtraum selber demonstrieren, da es für die nächste Gruppe und Euch als Dozierende eventuell unangenehm werden könnte.