

Diplomarbeit

Welpenüberwachung



Schule
Fachrichtung
Klasse
Autor
Ort, Datum

TEKO Schweizerische Fachschule Bern
Informatik Systemtechnik
B-TIS-17-T-a
Oliver Aegerter
Boll, 01.11.2020

Informationen zu dieser Dokumentation

Version	Datum	Autor	Änderungsgrund / Bemerkung
0.1	08.10.2020	Oliver Aegerter	Erstellung der Vorlage
0.2	08.10.2020	Oliver Aegerter	Ergänzung Phase Vorstudie
0.3	16.10.2020	Oliver Aegerter	Ergänzung Phase Grobkonzept
0.4	19.10.2020	Oliver Aegerter	Ergänzung Phase Detailkonzept
0.5	20.10.2020	Oliver Aegerter	Ergänzung Phase Realisierung
0.6	22.10.2020	Oliver Aegerter	Diverse Ergänzungen
1.0	23.10.2020	Oliver Aegerter	Fertigstellung des Dokuments

Beteiligte Personenkreis	
Benutzer / Anwender	Hermann Aegerter
Prüfung	Oliver Aegerter
Diplomlehrer	Xavier Pfaff
Fachexperte	Patrick Graber

Projektinformationen	
Auftraggeber	Hermann Aegerter
Vortragsname	Welpenüberwachung
Autor	Oliver Aegerter
Version	1.0

1 Management Summary

Die vorliegende Diplomarbeit beschäftigt sich mit der Evaluation, Umsetzung und Auswertung eines Überwachungssystems für Welpen innerhalb der Wurfbox. Das technologiebasierte Überwachungssystem hat zum Ziel die Züchter/-innen während den ersten Lebenstagen der Welpen zu entlasten. In dieser sensiblen Phase muss sichergestellt werden, dass die neugeborenen Tiere warm gehalten werden und eine konstante Gewichtszunahme verzeichnen. Insofern stellen die ersten Lebenstage für Züchter/-innen eine herausfordernde und kräftezehrende Phase dar, im Zuge derer sie die Welpen Tag und Nacht überwachen müssen.

Zielsetzung der vorliegenden Arbeit ist es, den Aufbau und die Inbetriebnahme eines technologiebasierten Überwachungssystems für Welpen exemplarisch aufzuzeigen. Die Konfigurationen des Systems lassen sich bedürfnisorientiert auf den Endnutzer ausrichten. Das Endresultat dieser Diplomarbeit ist ein Grundlagendokument, welches eine konkrete Einsatzmöglichkeit bei der Welpenaufzucht aufzeigt.

Anhand von Recherchen und im Austausch mit Fachleuten wurde ein mögliches Vorgehen festgelegt, erprobt, dokumentiert und evaluiert. Das methodische Vorgehen und die Orientierung an Projektphasen und Meilensteinen haben sich bei der konkreten Projektumsetzung als äusserst hilfreich herausgestellt. Im Zuge der Umsetzung kam es zu unterschiedlichen Herausforderungen, welche der Suche einer adäquaten Lösung bedurften. Ausserdem entstand die Haupte Erkenntnis, dass hinsichtlich des ständigen technologischen Fortschritts Lösungen für die flexible Kombination unterschiedlicher Komponenten gefunden werden müssen. Alles hat seine Zeit, daher muss man sich in Bezug auf Technologien und deren Einsatzmöglichkeiten ständig auf dem Laufenden halten und stets bemüht sein, sich diesbezüglich weiterzubilden. Aus der Diplomarbeit resultiert eine Standardlösung, welche exemplarisch, anhand eines konkreten Beispiels, die einzelnen Arbeitsschritte aufzeigt. Die Resultate und Erfahrungen eignen sich zum Wissenstransfer, ausserdem besticht die präsentierte Lösung durch ihre Erweiterbarkeit und Anpassungsfähigkeit auf verschiedene Kontexte und Bedürfnisse.

Inhaltsverzeichnis

1	Management Summary.....	3
2	Der Diplomand.....	9
2.1	Steckbrief Diplomand.....	9
2.2	Beruflicher Lebenslauf	9
3	Vorwort	11

I Phase Vorstudie

4	Aufgabenstellung	14
4.1	Projektauftrag	14
4.2	Auftraggeber	14
4.3	Projektleitung	14
4.4	Projektbeteiligte	15
4.5	Projekttitel und Projektkürzel	16
4.6	Zweck und Ziel des Auftrags.....	16
4.6.1	Persönliche Ziele	16
4.6.2	MUSS-Ziele	17
4.6.3	KANN-Ziele	18
4.7	Aufgabenabgrenzung	18
4.8	Risiken.....	19
4.8.1	Risikoeinflüsse und Beschreibung	19
4.8.2	Risikoanalyse	20
4.8.3	Massnahme	20
4.9	Abkürzungen	21
4.10	Glossar	22
4.11	Review Protokolle	24
4.11.1	Review 1 – Donnerstag 24.09.2020.....	24
4.11.2	Review 2 – Donnerstag 15.10.2020.....	25

5	Projekttermine	26
5.1	Projekttermine SOLL	26
5.2	Projekttermine IST	27
6	Anforderungen an das Projekt	29
6.1	Blockdiagramm Anforderung	29
6.2	Auswahl der physikalischen Komponenten	29

II Phase Grobkonzept

7	Auftrag	31
7.1	Bedürfnisse und Anforderungen an die Hauptkomponente	31
7.1.1	Mikrocontroller	31
7.1.2	Kamera	31
7.1.3	Übermittlungsgerät	32
7.1.4	Fazit Auftrag	32
7.2	Eignung	33
7.2.1	Evaluation Mikrocontroller	33
7.2.2	Entscheid Mikrocontroller	34
7.2.3	Beschreibung und Vorteile vom Arduino Uno	34
7.2.4	Evaluation Kamera	35
7.2.5	Entscheid Kamera	36
7.2.6	Beschreibung und Vorteile der Pixy Cam2	36
7.2.7	Evaluation Übermittlungsgerät	37
7.2.8	Beschreibung und Vorteile vom ipONEgprs	38
7.2.9	Fazit Hardware	38
7.3	Pilotierung	39
7.3.1	Einzelfunktion	39
7.3.2	IDE Programmierumgebung für Arduino	40
7.3.3	Einbindung Pixy Cam2 in das Arduino-Tool	41
7.3.4	Zusammenspiel	42

7.3.5	Fazit Pilotierung	42
7.4	Vorstudie Umsetzung	43
7.4.1	Kontextdiagramm Vorstudie Umsetzung.....	43
7.4.2	Blockdiagramm Vorstudie Umsetzung	44
7.4.3	Fazit Vorstudie Umsetzung.....	44
 III Phase Detailkonzept		
8	Aufbau Prototyp	46
8.1	Erweiterung Hardware	46
8.1.1	Erweiterung 4-Kanal Relais	46
8.1.2	Erweiterung ON-OFF Schalter	47
8.1.3	Fazit Erweiterung.....	47
8.2	Einbindung Übermittlungsgerät.....	48
8.2.1	Übermittlungsgerät Tool SysConf ipONE.....	48
8.2.2	SysConf ipONE Helpdesk.....	52
8.2.3	Fazit Übermittlungsgerät.....	53
8.3	Aufbau und Umsetzung Prototyp	54
8.3.1	Kontextdiagramm Prototyp	54
8.3.2	Blockdiagramm Prototyp.....	55
8.3.3	Der Aufbau	56
8.3.4	Programmierung und Software-Test	57
8.3.5	Fazit Aufbau Prototyp	57
8.4	Testkonzept.....	58
8.4.1	Testziele	58
8.4.2	Testrahmen	58
8.4.3	Testvorgehen.....	58
8.4.4	Fehlerklassifizierung	59
8.4.5	Testtabelle	60
8.4.6	Testplan.....	60

8.4.7	Fazit Testkonzept	60
8.5	Kosten	61
8.5.1	Fazit Kosten	61
 III Phase Realisierung		
9	Realisierung beim Kunden	63
9.1	Aufbau beim Kunden	63
9.1.1	Serielle Schnittstelle im IDE-Tool	64
9.1.2	Das Projekt	65
9.1.3	Stromlaufdiagramm des Projekts	67
9.1.4	Fazit Aufbau beim Kunden	67
9.2	Der Code	68
9.2.1	Fussdiagramm Software	68
9.2.2	Der programmierte Code	69
9.2.3	Fazit Code	73
9.3	Prüfung	74
9.3.1	Testprotokolle 1 MUSS-Ziele	74
9.3.2	Testprotokolle 1 KANN-Ziele	77
9.3.3	Prüfungsprotokoll 1	78
9.3.4	Nachprüfung	79
9.3.5	Testprotokolle 2 MUSS-Ziele	79
9.3.6	Prüfungsprotokoll 2	80
9.3.7	Fazit Prüfung	80
10	Quellenangabe und Literaturverzeichnis	81
10.1	Quellen aus dem Internet	81
10.2	Literaturverzeichnis	82
11	Abbildungsverzeichnis	83
12	Tabellenverzeichnis	85
13	Reflexion	86

13.1	Zielerreichung des Projekts	86
13.1.1	Zielsetzung des Projekts.....	86
13.1.2	Evaluation des Projekts	87
13.1.3	Grenzen und Chancen des Projekts	87
13.1.4	Fazit zum Projekt.....	88
13.2	Methodisches Vorgehen	89
13.2.1	Initialisierung des Projekts	89
13.2.2	Rückblick auf die Projektphasen	89
13.2.3	Zeitmanagement.....	90
13.2.4	Methodisches Fazit.....	90
13.3	Lessons learned	91
13.3.1	Persönliche Zielsetzung.....	91
13.3.2	Ausblick	92
13.3.3	Persönliches Fazit	92
13.4	Verdankung	93
13.5	Eidesstattliche Erklärung	94

2 Der Diplomand

Nachfolgend wird ein Steckbrief und ein kurzer beruflicher Lebenslauf des Diplomanden dargelegt.

2.1 Steckbrief Diplomand

Name	Aegerter
Vorname	Oliver Lars
Adresse	Gummenweg 3, 3067 Boll
Geburtsdatum	12.06.1984
Heimatort	Röthenbach i.E
E-Mail	oliver-aegerter@bluewin.ch
Telefonnummer	+41 79 935 99 61



2.2 Beruflicher Lebenslauf

Elektromonteur 2001-2014

Nach meiner Grundausbildung bei der Firma Muff+Schmutz AG in Bolligen, leistete ich 2005 den obligatorischen Militärdienst bei der Schweizer Armee. Nach Abschluss der Rekrutenschule konnte ich bei meinem Lehrbetrieb als Elektromonteur weiterarbeiten. Eine besondere Herausforderung war meine Mitarbeit beim Aufbau des neuen Firmenstandorts der Muff+Schmutz AG in Langenthal in den Jahren 2006-2007. Diese Erfahrung und die damit verbundene Verantwortungsübernahme sowie die enge Zusammenarbeit im Zweierteam waren für meinen beruflichen Lebenslauf von zentraler Bedeutung. Ende 2007 wechselte ich nach erfolgreicher Standortgründung zurück zur Niederlassung in Bolligen. Mit zunehmender Berufserfahrung konnte ich intern zum bauleitenden Monteur aufsteigen und leitete in der Folge Grossbaustellen. Dabei war ich für den reibungslosen Ablauf der Arbeiten auf den Baustellen zuständig, was unter anderem die Koordinierung der Mitarbeitenden, die Bereitstellung der Materialien, die weiteren Planungsschritte sowie die enge Zusammenarbeit mit der Bauherrschaft und den Bauplanern beinhaltete. Ausserdem war ich mit Teilen der Ausbildung der Lernenden beauftragt. Von 2011 bis 2013 bildete ich mich nebenberuflich weiter und erwarb ein Büro- und Handelsdiplom (Lehrgang Bürofachschule 1 und 2 mit DIPLOMA / ECDL).

Systemtechniker seit 2014

Im Jahr 2014 suchte ich eine neue berufliche Herausforderung, welche mir neue Perspektiven eröffnete. Diesen Anspruch konnte mir die Firma Securiton AG in Zollikofen erfüllen, wo ich seit dem Sommer 2014 arbeite. Das global operierende Unternehmen stellt einen interessanten und reizvollen Kontrast zum mittelständigen Unternehmen dar, wo ich bis anhin arbeitete. Zu meinen grundlegenden Tätigkeiten als Techniker im Kundendienst gehören Wartungen, Störungsbehebungen und Umbauten von Brand- und Einbruchsmeldeanlagen. Mit der zunehmenden Digitalisierung fallen bei der Ausübung des Berufs verschieden Programmier- und Kontrollarbeiten an, dabei ist es besonders interessant, die Auswirkungen der Digitalisierung und den damit verbundenen Wandel zu begleiten und firmenintern mitzugestalten. Ausserdem verlangt der Beruf ausgeprägte Fähigkeiten im Kontakt und Umgang mit den Kunden. Im Jahr 2018 wurde ich zum Fachtechniker im Kundendienst befördert, seither bin ich für den Support und Betrieb von Sonderbrandsystemen im Kanton Bern sowie in der Westschweiz verantwortlich. Im Herbst 2017 begann ich mit der Weiterbildung zum Techniker HF Informatik (Systemtechnik) an der TEKO (Schweizerische Fachschule), der Abschluss ist für das Herbstsemester 2020 vorgesehen.

3 Vorwort

Die vorliegende Diplomarbeit entsteht im Rahmen des Studiums an der höheren Fachschule bei der TEKO. Es handelt sich dabei um eine Qualifikationsarbeit am Ende der Weiterbildung zum Techniker HF Informatik (Systemtechnik). Im Zuge des Studiums hatten die Studierenden die Möglichkeit ihr fachspezifisches Wissen zu vertiefen und zu erweitern. Das erarbeitete Wissen und die erlangten Fähigkeiten wurden in verschiedenen Semesterarbeiten unter Beweis gestellt. Diese Semesterarbeiten hatten zum Ziel, die Studierenden mit der Erarbeitung, dem Aufbau, der Durchführung und Evaluierung einer Arbeit vertraut zu machen. Die Diplomarbeit bildet den Abschluss des Studiums, ihr Hauptziel ist es, eine spezifische Aufgabenstellung innerhalb einer vorgeschriebenen Zeitspanne selbstständig, umfassend und zweckmässig zu lösen. Dabei wird ein besonderes Augenmerk auf die Dokumentation und Präsentation gelegt.

Die Vorgabezeit für die Diplomarbeit beträgt sechs Wochen. In dieser Zeit finden keine Präsenzveranstaltungen statt, sodass neben der hauptberuflichen Tätigkeit genügend zeitliche Ressourcen für die Umsetzung der Diplomarbeit zur Verfügung stehen. Am 18. September 2020 begann die offizielle Umsetzungsphase für die Diplomarbeit; die Abgabe ist auf den 02. November 2020 terminiert.

Die Arbeit orientiert sich am Basis-Phasenkonzept, welches verschiedene Phasen für ein zielgerichtetes und effizientes Projektmanagement vorsieht. Das methodische Vorgehen führt zu mehr Transparenz und unterstützt dabei die Projektplanung und -steuerung:

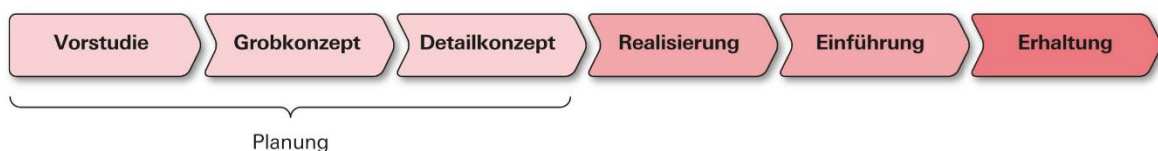


Abbildung 1: Basis-Phasenkonzept

Im Rahmen der vorliegenden Diplomarbeit werden die Phasen *Vorstudie*, *Grobkonzept*, *Detailkonzept* sowie die *Realisierung* fokussiert und detailliert ausgeführt. Aus der Arbeit resultiert ein Grundlagendokument, welches die *Einführung* und *Erhaltung* des beschriebenen Systems bedürfnisgerecht und in unterschiedlichen Kontexten erleichtert. Daher sind diese Phasen nicht Teil der Zielsetzung der vorliegenden Arbeit und werden ausgeklammert.

Nachfolgend werden die Phasen erläutert und die zentralen Merkmale dargelegt.

Vorstudie (Kapitel 4-6): Die Vorstudie dient der grundsätzlichen Klärung zentraler Informationen und der Situation. Dabei werden die Rollen der verschiedenen involvierten Akteure sowie die Bedürfnisse des Auftraggebers geklärt. Dabei werden Ziele und Risiken abgewogen. Des Weiteren wird die Projektplanung inklusive der Projekttermine ausgewiesen.

Grobkonzept (Kapitel 7): Das Grobkonzept beinhaltet eine erste Evaluation des Vorhabens, auf der in der Folge das Detailkonzept aufbauen kann. Im Abschnitt Grobkonzept werden die Bedürfnisse und Anforderungen der Systemkomponenten evaluiert, verglichen und deren Eignung abgeschätzt. Ausserdem findet die Pilotierung der Einzelkomponenten sowie des Zusammenspiels der Hardware statt. Im Rahmen des Grobkonzepts wird zudem ein Konzept für die Umsetzung der Vorstudie erarbeitet.

Detailkonzept (Kapitel 8): Das Detailkonzept konkretisiert die Pläne hinsichtlich der Projektumsetzung. Im Zuge dieser Projektphase werden zusätzliche Komponenten evaluiert und ein Prototyp des Systems umgesetzt. Das Testkonzept wird definiert sowie die Projektkosten aufgezeigt.

Realisierung (Kapitel 9): Die Realisierung beschreibt die Umsetzung der Pläne und Konzepte. Im Rahmen dieser Projektphase wird die Welpenüberwachung beim Kunden aufgebaut und geprüft. Dabei werden die Installationen und Programmierungen im Detail beschrieben und aufgezeigt.

Oliver Aegerter

Höhere Fachschule, TEKO

01.11.2020

I Phase Vorstudie

4 Aufgabenstellung

Die nachfolgenden Kapitel beschreiben die grundlegende Ausgangslage und Zielsetzung des Projekts. Dabei wird eine Risikoanalyse und Projektabgrenzung vorgenommen. Ausserdem werden die Abkürzungen erläutert und ein Glossar bereitgestellt.

4.1 Projektauftrag

Im Rahmen des Projekts soll ein Überwachungssystem entwickelt werden, welches die Überwachung von Welpen in der Wurfbox erleichtert. Dies ist insbesondere in den ersten Lebenstagen von Welpen sehr wichtig und stellt eine gute Entwicklung der Welpen sicher. Beim Ausbruch eines Welpen aus der Wurfbox soll das System den Züchter bzw. die Züchterin per SMS alarmieren. Eine wichtige Spezifikation ist die Blockierung des Alarms in bestimmten Situationen, beispielsweise, wenn die Welpen bewusst aus der Box herausgenommen werden. Das Projekt soll eine bedürfnisgerecht auf den Kunden abgestimmte Lösung aufzeigen, aber auch als Basis für Erweiterungen dienen. So ist das Resultat dieser Diplomarbeit je nach Wunsch mit weiteren Komponenten erweiterbar und kann mit weiteren Alarmierungsmitteln ausgestattet werden.

4.2 Auftraggeber

Der Auftraggeber ist ein Züchter für Brandlbracken: Hermann Aegerter erteilte mir den Auftrag, ein Überwachungssystem für Welpen zu planen, umzusetzen und zu prüfen.

4.3 Projektleitung

Der Auftraggeber ist Hermann Aegerter, er ist gleichzeitig Besitzer der Welpen. Die Projektleitung und Realisierung des Projekts obliegt Oliver Aegerter. Xavier Pfaff begleitet den Diplomanden durch das Projekt. Als Fachexperte fungiert Patrick Graber.

4.4 Projektbeteiligte

Die Initialisierung, Konzeption und Realisierung des Projekts fällt in den Zuständigkeitsbereich von Oliver Aegerter. Als Projektleiter übernimmt er die Gesamtverantwortung des Projektes und etabliert eine Zusammenarbeit mit Xavier Pfaff, welcher das Projekt als Diplomlehrer begleitet. Patrick Graber wird an der Präsentation als Fachexperte teilnehmen.

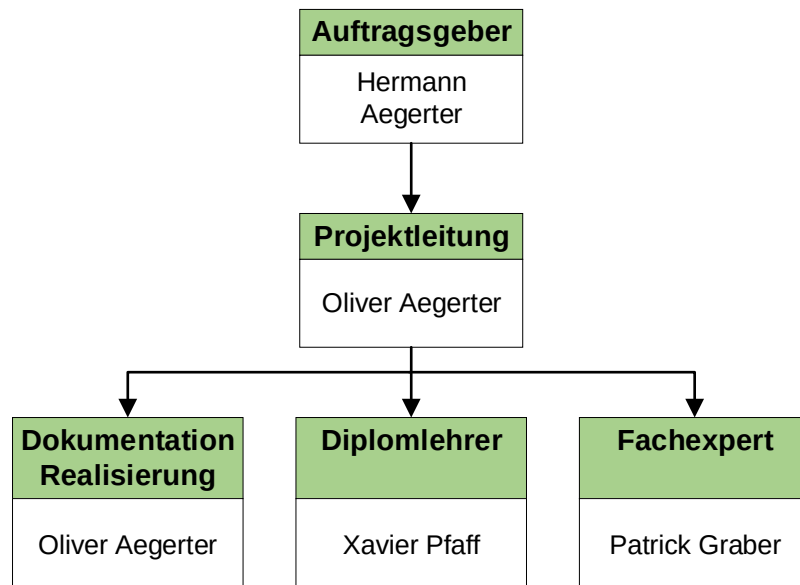


Abbildung 2: Projektbeteiligte

4.5 Projekttitle und Projektkürzel

Der Projekttitle lautet Welpenüberwachung, da sich das Projekt mit der Überwachung von Welpen befasst und einen Basisvorschlag für ein technologiebasiertes Überwachungssystem ermittelt. Daraus leitet sich das Projektkürzel WÜ ab.

4.6 Zweck und Ziel des Auftrags

Das folgende Unterkapitel definiert die persönlichen sowie die projektbezogenen *MUSS*- und *KANN*-Ziele und nimmt eine grundlegende Abgrenzung der Arbeit vor.

4.6.1 Persönliche Ziele

In den sechs Semestern an der TEKO hat mir am besten das Fach *Embedded Systeme* gefallen. Umso mehr freut es mich, dass ich in meine Diplomarbeit das Erlernte in diesem Fach anwenden und umsetzen kann. Nachfolgend erläutere ich meine persönliche Ziele im Zusammenhang mit meiner Diplomarbeit.

- **Bessere Kenntnisse der Hardware:**
 - Beim Umgang mit Hardware verfüge ich bereits über viel Erfahrung, gleichzeitig verspreche ich mir bei der Realisierung des Projekts mein Wissen zu vertiefen und eine gewinnbringende Auseinandersetzung mit den Systemkomponenten.
- **Bessere Kenntnisse der Software:**
 - Die Programmiersprache C wird im Bereich der Softwareprogrammierung eine grosse Herausforderung darstellen. Berufliche Erfahrungen damit habe ich noch keine, ich habe sie lediglich im Zuge meines Studiums kennenlernen dürfen.
- **Kundenzufriedenheit und Nützlichkeit des Systems:**
 - Der Kunde kann das System nutzen und es stellt eine echte Erleichterung bei der Überwachung der Welpen dar.

4.6.2 MUSS-Ziele

Mit dem Kunden wurden folgende Ziele besprochen, sie müssen im vorgeschlagenen System umgesetzt werden:

- **Dauerhafte Überwachung der Welpen:**
 - Das System muss Tag und Nacht einsetzbar sein.
- **Alarmierung erfolgt über ein Übermittlungsgerät:**
 - Bei einem Ausbruch eines oder mehrerer Welpen wird der Kunde mit einer SMS benachrichtigt.
- **Fehlerfreier Betrieb über die gewünschte Überwachungszeit der Welpen:**
 - Störquellen sind zu erkennen und zu beheben.
- **System muss an- und abschaltbar sein:**
 - Die Welpen sind nicht immer in der Wurfbox. Die Alarmierung muss daher blockierbar sein.

4.6.3 KANN-Ziele

Mit dem Kunden wurden folgende Ziele besprochen, sie können im vorgeschlagenen System umgesetzt werden:

- **Mit Hilfe von LEDs kann der Zustand des Projekt erkannt werden:**
 - Mit LEDs wird die Anzahl der Welpen angezeigt und es ist sichtbar, ob der Schalter auf On/Off gestellt ist.
- **Optische Alarmierung:**
 - Zusätzliche Alarmierungsmittel, wie zum Beispiel Blitzleuchten oder Sirenen, können integriert werden.
- **Temperaturüberwachung:**
 - Für die Welpen ist eine konstante Temperatur sehr wichtig. Das System misst die Temperatur und sendet eine SMS bei Temperaturschwankungen.
- **System überall einsetzbar:**
 - Das Projekt soll so gebaut werden, dass das System leicht demontiert, an einem neuen Standort wieder montiert und in Betrieb genommen werden kann.

4.7 Aufgabenabgrenzung

Das System zur Welpenüberwachung soll gemäss den Kundenbedürfnissen umgesetzt werden. Da es ausserdem explizites Ziel der vorliegenden Diplomarbeit ist, ein Grundlagentokument zu schaffen, dessen beschriebenes System auf verschiedene Kontexte angewendet werden und modifiziert werden kann, stellt das konkret realisierte System einen Prototypen dar. Insofern spielt der Aspekt der Optik eine untergeordnete Rolle, später kann das System in einem Gehäuse integriert werden. Für das Projekt wird das Gesamtsystem auf einem Holzbrett montiert und für Testzwecke mit einem Teleskoparm an die Wand geschraubt. Das System ist so einsatzbereit und kann betrieben werden.

4.8 Risiken

Bei jedem Projekt gibt es Risiken, die man berücksichtigen muss. Nachfolgend werden zentrale Risiken genannt und mögliche Massnahmen zusammengestellt.

4.8.1 Risikoeinflüsse und Beschreibung

Tabelle 1: Risikoeinflüsse und Beschreibung

Nr.	Risiko	Beschreibung
1	Zeitaufwand wird zu gross sein	Die Programmierung beansprucht zu viel Zeit.
2	Hardware ist nicht kompatibel	Die Hardware wird von verschiedenen Herstellern bezogen. Hier stellt sich immer die Frage, ob die Geräte kompatibel sind.
3	Zeitaufwand der Dokumentation wird zu gross sein	Der Zeitaufwand ist zum jetzigen Zeit noch nicht absehbar.
4	Materialkosten zu hoch	Das angestrebte Budget wird überzogen.
5	Installation stört die Hündin	Es besteht die Gefahr, dass sich die Hündin durch die Installation und die Grösse des Projekts gestört fühlt und Stress erlebt. Dies gilt es zu vermeiden.

4.8.2 Risikoanalyse

Die Risikoanalyse liefert wichtige Informationen für die Bewertung meines Projekts. Hier werden die Risiken in *Eintrittswahrscheinlichkeit* und *Tragweite* eingeteilt und bewertet.

Tabelle 2: Risikoanalyse

Eintrittswahrscheinlichkeit	Hoch	1		
	Mittel	2/4	3	
	Gering		5	
		Gering	Mittel	Hoch
		Tragweite		

4.8.3 Massnahme

Tabelle 3: Massnahme

Nr.	Risiko	Massnahme
1	Zeitaufwand wird zu gross sein	Vorbereitend muss ein gutes Zeitmanagement geplant werden, ausserdem muss die Programmiersprache C erlernt werden.
2	Hardware ist nicht kompatibel	In der Pilotierung (Phase Grobkonzept) muss dieser Aspekt berücksichtigt werden und allenfalls auf andere Hardware zurückgegriffen werden.
3	Zeitaufwand der Dokumentation wird zu gross sein	Sich strikt an die Zeitplanung halten und die Meilensteine berücksichtigen.
4	Materialkosten zu hoch	Hardware vergleichen und die effizienteste Variante wählen.
5	Installation stört die Hündin	Die Installation so hoch wie möglich montieren und Geräuschemissionen vermeiden.

4.9 Abkürzungen

Folgende Abkürzungen finden in der Arbeit Verwendung und werden hiermit kurz erläutert:

AC	Alternating Current, Wechselstrom
DC	Direct Current, Gleichstrom
EEPROM	Electrically Erasable Programmable Read-Only Memory
gprs	General Packet Radio Service
HW	Hardware
ICSP	In Circuit Serial Programming
IP	Internet Protocol, eine Adresse im Computernetz
I²C	Inter-Integrated Circuit
KI	Künstliche Intelligenz
LED	Light-emitting diode, Leuchtdiode
lte	Long Term Evolution
PIN	Persönliche Identifikationsnummer
PWM	Pulse Width Modulation, Pulsdauermodulation
RAM	Random Access Memory
SMS	Short Message Service, Kurznachrichtendienst
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory, statisches RAM
SW	Software
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

4.10 Glossar

Algorithmus:

Wird in Computerprogrammen verwendet und besteht aus zahlreichen, definierten Einzelschritten. Sie definieren Handlungsvorschriften, die zur Lösung von Problemen oder Klassen eingesetzt werden.

Bootloader:

Spezielle Software, die das auf dem Computer installierte Betriebssystem in den Arbeitsspeicher lädt. Auch Startprogramm genannt.

Button:

Schalter für etwas ein- und ausschalten zu können.

Brandlbracke:

Mittelgrosse Jagdhundrasse aus Österreich, welche vielseitig in der Jagd einsetzbar ist.

Dual Core:

Ist ein Mehrkernprozessor, der mit der entsprechenden Software eine Steigerung der Gesamtrechnerleistung ohne Erhöhung der Taktfrequenz ermöglicht.

Embedded Systeme:

Ein eingebettetes System mit Prozessoren, Stromversorgung, Speicher und externen Schnittstellen.

Ethernet:

Ermöglicht den Datenaustausch zwischen Hardware und Software über ein Kabel.

Flash Memory:

Ist ein digitaler Speicher für eine nichtflüchtige Speicherung, ohne Erhaltung- und Energieverbrauch. Genaue Bezeichnung ist Flash-EEPROM.

Hardware:

Elektronisches und mechanisches Bauteil eines datenverarbeitenden Systems.

Mikrocontroller:

Ist ein Halbleiterchip, der einen Prozessor (Arbeits- und Programmspeicher) hat. Mikrocontroller haben Peripheriefunktion wie z.B. USB, I²C, SPI, serielle Schnittstellen und Ethernet-Schnittstellen.

Open Source:

Ist eine öffentliche Software, deren Quelltext von Dritten genutzt, geändert und eingesehen werden kann.

Pulsdauermodulation (PWM):

Ist eine technische Grösse die zwischen zwei Werten wechselt. Dabei wird bei einer konstanten Frequenz der Tastgrad moduliert, also die Dauer eines Rechteckpulses moduliert.

Relais:

Das Relais wird über einen Steuerstromkreis aktiviert und durch das elektromagnetische Wirken der Spule, wird der Schaltkreis geschlossen oder geöffnet.

Review:

Ein Review ist eine systematische Übersichtsarbeit, die den Forschungsstand zu einem Thema erläutert.

Software:

Die Software bestimmt, was ein softwaregesteuertes und elektronisches Bauteil tun soll.

SysConf:

Ist eine Software zum Programmieren des Übermittlungsgeräts.

Tool:

Ist ein Hilfswerkzeug mit dem der Computer oder eine Applikation (App) verwaltet werden kann.

Wurfbox:

Auch Wurfkiste genannt, ist eine hölzerne Kiste, in der die Welpen ihre ersten Lebenswochen verbringen.

4.11 Review Protokolle

Für dieses Projekt sind zwei Reviews mit meinem Diplomlehrer Xavier Pfaff geplant. Ziel dabei ist es, offene Fragen und das weitere Vorgehen zu besprechen.

4.11.1 Review 1 – Donnerstag 24.09.2020

Tabelle 4: Review 1

Tätigkeit	Person	Zeit geplant	Zeit effektiv
Projekttermine (Meilensteine)	AeO	3 h	3 h
Material bestellen	AeO	4 h	5 h
Verknüpfungen Arduino und Pixy Cam	Aeo	5 h	5 h
Programmiersprache C	AeO	5 h	8 h
Vorlagen für Dokumentation erstellen	AeO	2 h	1 h
Total		19 h	22 h
Probleme und Hilfestellung			
- Die Bibliothek für die Pixy Cam2 und Arduino hatte ich schnell gefunden und im Tool für das Programmieren integriert. Die ersten Versuche waren jedoch zeitintensiv. - Die Analyse der Hardware beansprucht mehr Zeit als geplant. - Die Vorstudie mit der Programmiersprache C beansprucht ebenfalls mehr Zeit als eingeplant.			
Massnahme			
Bei Problemen und Stolpersteinen muss eine Klärung der Aufgabe durchgeführt werden; hierbei gilt es, sich die theoretischen Grundlagen zu erarbeiten und anschliessend auf das Projekt anzuwenden. Diese Auseinandersetzung und der anschliessende Transfer müssen sorgfältig und konzentriert stattfinden.			
Reflektion			
Der aufgrund der Probleme resultierende Zeitverlust konnte mit erhöhtem Engagement und den Tipps des Diplomlehrers aufgeholt werden. Somit befinde ich mich wieder im Zeitplan.			
Einen ersten Aufbau und die Verknüpfung konnte ich mit Hilfe von diversen Hilfsmitteln (z.B. Tutorials) selber umsetzen und konfigurieren.			
Nächste Schritte			
- Vorstudie Programmiersprache C - Montage und Inbetriebnahme beim Kunden			

4.11.2 Review 2 – Donnerstag 15.10.2020

Tabelle 5: Review 2

Tätigkeit	Person	Zeit geplant	Zeit effektiv
Erstellung Testkonzept	AeO	2	2
Inbetriebnahme des Systems	Aeo	5	4
Testen	AeO	6	11
Dokumentation zu ca. 75% erstellt	AeO	40	39
Vorbereitung der Online-Publikation	AeO	10	8
Total		63	64
Probleme und Hilfestellung			
• Es stellte sich mir die Frage, wie die Dokumentation zielführend gestaltet werden soll. • Ausserdem bestanden Unklarheiten bezüglich der Anzahl und Art der Schemata, welche der Visualisierung meiner Arbeit dienen. • Zudem bedurfte der Inhalt der Online-Publikation einer Klärung. Insbesondere, ob alle technischen Dokumente separat aufgezeigt werden müssen oder das Hochladen der Dokumentation ausreichend ist.			
Massnahme			
In der Auseinandersetzung mit den Schulungsunterlagen bot sich das Basis-Phasenkonzept besonders an, damit kann innerhalb der Dokumentation das methodische Vorgehen transparent gemacht werden und so die Projektplanung erleichtern. Daher entschied ich mich, meine Arbeit gemäss der dort beschriebenen Phasen aufzubauen. In der Diskussion mit meinem Diplomlehrer während dem 2. Review tauschte ich mich über mögliche Visualisierungsstrategien aus und konnte mir so eine bessere Vorstellung bezüglich nützlicher Schemata machen. Ausserdem habe ich mich in den Unterlagen zu den Rahmenbedingungen der vorliegenden Diplomarbeit über die Inhalte der Online-Publikation informiert.			
Reflektion			
Das Basis-Phasenkonzept, welches wir während der Ausbildung theoretisch kennengelernt haben, lässt sich nun im Rahmen dieses Projekts praktisch umsetzen. Das erarbeitete Wissen erleichtert mir somit die Arbeit und hat einen echten Nutzen. Ausserdem ist es sinnvoll sich zu Beginn einer Arbeit eingehend mit den Vorgaben und Rahmenbedingungen auseinanderzusetzen und sie bereits bei der Konzipierung zu berücksichtigen.			
Nächste Schritte			
• Erörterung der Inhalte, welche in den jeweiligen Phasen beschrieben werden sollen • Erstellung nützlicher Schemata zur Visualisierung der Arbeiten • Schreiben der Dokumentation • Formatierung und Fertigstellung der Arbeit			

5 Projekttermine

Das nachfolgende Kapitel gibt einen Überblick über die zeitliche Struktur des Projekts. Im Zuge der Planung wurden Projekttermine festgelegt und ihre Einhaltung dokumentiert.

5.1 Projekttermine SOLL

Für meine Diplomarbeit habe ich mir wichtige Projekttermine und Meilensteine definiert. Für das Projektmanagement sind Meilensteine besonders zentral, sie definieren Etappen, welche anhand von Zwischenzielen überprüft werden können. Die Meilensteine fallen auf die Termine für das 1. und 2. Review, wo in der Auseinandersetzung mit dem Diplomlehrer Zielsetzungen retrospektiv und prospektiv besprochen werden. Ausserdem bildet der Abgabetermin den finalen Meilenstein, hier findet im Vorfeld eine letzte Zielüberprüfung statt.

18.09.2020	Start Diplomarbeit
21.09.2020	Projektstruktur definieren
24.09.2020	1. Review
28.09.2020	Vorstudie Hardware und Software
29.09.2020	Material bestellen
06.10.2020	Bei Kunden montieren
06.10.2020	Inbetriebnahme
07.10.2020	Testen
08.10.2020	Dossier erstellen
15.10.2020	2. Review
22.10.2020	Management Summary erstellen
23.10.2020	Dossier beenden
26.10.2020	Online-Publikation erstellen
01.11.2020	Abgabe Diplomarbeit
02.11.2020	Präsentation erstellen
13.11.2020	Ende der Diplomarbeit

Abbildung 3: Projekttermine SOLL

5.2 Projekttermine IST

Tabelle 6: Projekttermine IST

Datum	Thema Beschreibung	Aufwand in h
Initialisierung		
18.09.2020	Start Diplomarbeit	3
21.09.2020	Projektstruktur definieren	6
23.09.2020	Vorbereitung Review 1	3
24.09.2020	Durchführung Review 1	2
24.09.2020	Zusammenfassung Review 1	2
Vorstudie		
25.09.2020	Projektauftrag definieren	2
25.09.2020	Zweck und Ziele definieren	2
25.09.2020	Abgrenzungen definieren	2
25.09.2020	Risiko-Evaluierung	2
26.09.2020	Projekttermine definieren	5
26.09.2020	Anforderungen an das Projekt klären	3
Grobkonzept		
28.09.2020	Bedürfnisse und Anforderungen Hauptkomponente	6
28.09.2020	Vorstudie Hardware und Software	3
29.09.2020	Eignung Hardware definieren	5
29.09.2020	Entscheidung für Hardware- Komponenten	2
29.09.2020	Material bestellen	2
30.09.2020	Pilotierung	6
01.10.2020	Einzelfunktionen testen	3
01.10.2020	Zusammenspiel testen	3
01.10.2020	Vorstudie Umsetzung	2
Detaillkonzept		
02.10.2020	Erweiterung Hardware	4
02.10.2020	Zusätzliches Material bestellen	2
03.10.2020	Einbindung Übermittlungsgerät	8
05.10.2020	Aufbau Prototyp	5
05.10.2020	Testkonzept erstellen	2
05.10.2020	Kosten zusammenstellen	2

Realisierung		
06.10.2020	Projekt beim Kunden aufbauen	4
06.10.2020	Inbetriebnahme	4
07.10.2020	Projekt prüfen	8
08.10.2020	Nachprüfung	3
08.10.2020	Dossier erstellen	1
08.10.2020	Dokumentation Vorstudie	10
09.10.2020	Besprechung beim Kunden	4
14.10.2020	Vorbereitung Review 2	3
15.10.2020	Durchführung Review 2	2
15.10.2020	Zusammenfassung Review 2	2
16.10.2020	Dokumentation Grobkonzept	20
19.10.2020	Dokumentation Detailkonzept	8
20.10.2020	Dokumentation Realisierung	18
22.10.2020	Management Summary erstellen	4
23.10.2020	Dossier abschliessen und überarbeiten	15
26.10.2020	Online-Publikation erstellen	8
27.10.2020	Online-Publikation kontrollieren	3
28.10.2020	Online-Publikation abschliessen	2
01.11.2020	Online-Publikation Freigabe	1
01.11.2020	Dokumentation Freigabe	2
Präsentation		
02.11.2020	Präsentation erstellen	6
03.11.2020	Präsentation kontrollieren	3
04.11.2020	Präsentation abschliessen	2
13.11.2020	Durchführung Präsentation	1
13.11.2020	Ende der Diplomarbeit	1
Total Aufwand in Stunden		222

6 Anforderungen an das Projekt

Im Kundengespräch konnten die gewünschten Eigenschaften der Welpenüberwachung evaluiert und ein mögliches System vorgeschlagen werden. Diese Klärung ermöglicht es, die Anforderungen an das Projekt zu erkennen und die weiteren Planungsschritte in Angriff zu nehmen.

6.1 Blockdiagramm Anforderung

Nachfolgendes Blockdiagramm erlaubt eine erste grobe Übersicht, daraus geht hervor, welche Hardware-Komponenten ich benötige.

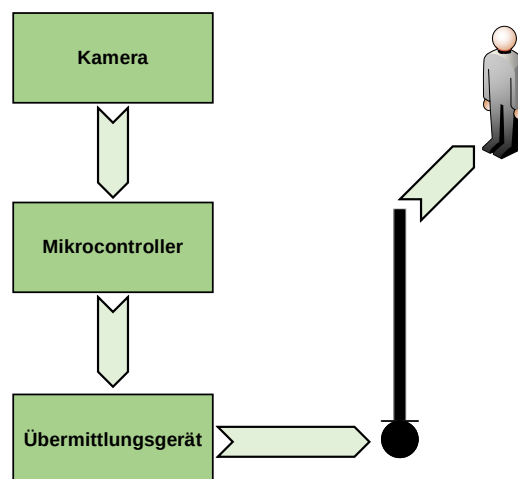


Abbildung 4: Blockdiagramm Anforderung

6.2 Auswahl der physikalischen Komponenten

Nach der Erstellung des Blockdiagramms erkannte ich, dass ich eine Kamera, einen Mikrocontroller und ein Übermittlungsgerät benötige. Im Fach Embedded Systeme haben wir mit Arduino gearbeitet. Ich kann mir vorstellen, dass ich mein Projektvorhaben mit Arduino umsetzen werde, da ich diese Hardware-Komponente eingehend kenne und schon praktische Erfahrungen mit diesen Komponenten habe. Die Auswahl an Arduino Shields, Sensoren und Zubehör ist sehr gross und auch für den Kunden sehr kostenfreundlich.

II Phase Grobkonzept

7 Auftrag

Im Kapitel Auftrag werde ich zunächst die Bedürfnisse und Anforderung an das System klären und eine Evaluation der Hauptkomponente durchführen.

7.1 Bedürfnisse und Anforderungen an die Hauptkomponente

Die nachfolgende Auflistung ist das Resultat eingehender Überlegungen zu spezifischen Möglichkeiten der Hardware-Komponenten. Im Hinblick auf das Gesamtsystem müssen die einzelnen Komponenten folgende Eigenschaften erfüllen:

7.1.1 Mikrocontroller

- Empfangen und Verarbeiten von Signalen
 - digital und analog
- Weiterleitung von Signalen
- Genügend Rechnerleistung
- Serielle Kommunikation
- Genügend Ein- und Ausgänge

7.1.2 Kamera

- Schnelle Objekterkennung
 - Unterscheidung von Objekten
- Schnelle Weiterleitung der Daten
- Vielseitige Montage

7.1.3 Übermittlungsgerät

Beim Übermittlungsgerät konnte ich auf ein Produkt meiner Firma zurückgreifen, die mir ein ipONEgprs zur Verfügung stellt. Das Gerät eignet sich für die Versendung von Sprachmitteilungen. Es ist aber speziell darauf hinzuweisen, dass ein Übermittlungsgerät mit gprs nur eine kurze Lebensdauer hat. Der Grund dafür ist, dass die Swisscom AG, angekündigt hat, das 2G-Netz abzustellen. Folglich müssen alle Geräte, die mit 2G-Technologie ausgerüstet sind, durch ipONElte ausgetauscht werden. Ich empfehle für zukünftige Wahl der Übermittlungsgeräte auf ein Gerät dieser Technologie zurückzugreifen.

7.1.4 Fazit Auftrag

Die Hardware muss für das spätere Zusammenspiel im Gesamtsystem kompatibel sein. In einem nächsten Schritt ist es wichtig, entsprechend der Anforderung eine optimale Hardware-Lösung zu finden. Im Kapitel 7.2 skizziere ich die Eignungsanalysen für die Hardware der Welpenüberwachung.

7.2 Eignung

Bei der Eignungsprüfung geht es darum, das effizienteste Hardware-Produkt für das Projekt zu finden. Hierfür wird ein Vergleich zweier Hardware-Produkte angestellt, welche für das Vorhaben in Frage kommen.

7.2.1 Evaluation Mikrocontroller

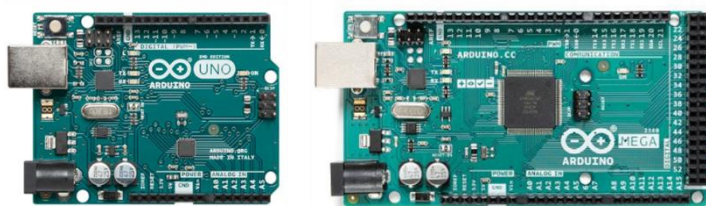


Tabelle 7: Evaluation Mikrocontroller

Abbildung 5: Arduino Uno & Arduino Mega

Typ	Arduino Uno	Arduino Mega
Takt	16 MHz	16MHz
Bit	8	8
Ausführung		
Mikrocontroller	ATmega328	ATmega2560
Flash Memory	32 KB	256 KB
SRAM	2KB	8 KB
EEPROM	1	4
Elektronische Werte		
Spannungseingang	9V DC	9V DC
Spannung	5V	5 V
Anschlüsse		
Analogeingänge	6	16
Digital I/O Pins	14	54
Digital I/O mit PWM	6	14
USB	ja	ja
ICSP	ja	ja
Allgemein		
Masse L x B	68.6mm x 53.4mm	101.52mm x 53.3mm
Preis	CHF 27.90	CHF 34.35

7.2.2 Entscheid Mikrocontroller

Beide Mikrocontroller eignen sich für meine Diplomarbeit. Für das Vorhaben reichen die Eigenschaften des Arduino Uno aus. Dieser hat zum Vorteil, dass er kleiner und kostengünstiger ist als der Arduino Mega. Ausserdem werden für das Projekt nicht so viele digitale und analoge Anschlüsse benötigt.

7.2.3 Beschreibung und Vorteile vom Arduino Uno

Bei Arduino handelt es sich einerseits um einen Hardware-Hersteller und andererseits um eine Software, welche Open Source ist. Die Hardware besteht aus einem E/A-Bord mit einem Mikrocontroller sowie digitalen und analogen Ein- und Ausgängen. Die Arduino Entwicklungsumgebung soll auch technisch weniger versierten Personen den Zugang zur Programmierung und zu Mikrocontrollern ermöglichen. Die Programmiersprache ist auf C beziehungsweise C++ aufgebaut. Die Programmierung wird mit zahlreichen Bibliotheken und Beispielen vereinfacht. Die Verwendung von Arduino kann sehr vielfältig sein. Von einfachen Sensorauswertungen bis zu komplexen Steuerungen sind keine Grenzen gesetzt. Mit dem vorprogrammierten Bootloader kann man direkt über die serielle Schnittstelle programmieren. Mit dem ATmega8U2 wird das USB-Signal in ein serielles Signal umgesetzt. Ausserdem ist auf der Platine eine 6-polige ISP-Schnittstelle und diese ermöglicht es, direkt zu programmieren. Beim Arduino Uno sind 14 digitale Pins verfügbar. Von diesen 14 Pins können 6 für die Pulsdauermodulation verwendet werden. Weiter stehen noch 6 analoge Pins zur Verfügung. Es gibt die Möglichkeit, Zusatzplatinen auf die Arduinos aufzustecken. Zum Beispiel gibt es die Möglichkeit, eine Ethernet-Platine aufzustecken und so wird eine Kommunikation über ein Netzwerk ermöglicht. Die Spannungsversorgung erfolgt über den USB- Anschluss oder über die Eingangsbuchse.

7.2.4 Evaluation Kamera



Tabelle 8: Evaluation Kamera

Abbildung 6: Pixy Cam2 & M5 Camera OV 2640

Typ	Pixy Cam2	M5 Camera OV 2640
Takt	204 MHz	240MHz
Ausführung		
Prozessor	NXP LPC4330 Dual Core	Tensilica LX6 Dual Core
Flash Memory	2 Mbytes	4 Mbytes
Speicher	264 Kbytes RAM	520 Kbytes SRAM
Elektronische Werte		
Spannungseingang	5V	5V
Anschlüsse		
Schnittstellen	UART / SPI / I ² C, USB	UART / USB / GROVE / I ² C / PWM
Allgemein		
Bildauflösung	1296 x 976 mit integriertem Bildflussprozessor	1600 x 1200
Pixel	1.3	2
Objekterkennung	Ja	Ja
Bild pro Sekunde	60	5 - 6
Sichtfeld	60 Grad horizontal und 40 Grad vertikal	65 Grad horizontal
Lichtquellen	Ja, 20 Lumen	Nein
Arduino kompatibel	Ja	Ja
Masse L x B	38.1mm x 41.9mm	40mm x 48mm
Preis	CHF 77.50	CHF 27.90

7.2.5 Entscheid Kamera

Die Kameras haben beide ihre Vor- und Nachteile. Unter Zuhilfenahme verschiedener Produktrezensionen (Tools und Videos) fiel der Entscheid auf die Pixy-Cam2. Besonders überzeugend sind die 60 Bilder pro Sekunde sowie die integrierte Lichtquelle.

7.2.6 Beschreibung und Vorteile der Pixy Cam2

Die Pixy Cam2 kann unterschiedliche Objekte erkennen. Durch direktes Betätigen des Buttons vor Ort oder im Software-Tool erlernt die Kamera Objekte zu erkennen und zu unterscheiden. Anders als ihre Vorgänger, verfügt die Pixy Cam2 über einen neuen Algorithmus. Sie ist in der Lage, nach vorne zu schauen und kann Linien folgen oder Kreuzungen erkennen. Des Weiteren kann sie Objekte, wie zum Beispiel Strassenschilder, erkennen und interpretieren. Man kann ihr beibringen, dass beispielsweise ein nach links zeigender Pfeil ein abdrehen nach links bedeutet. Sie verarbeitet 60 Bilder in der Sekunde, das macht sie sehr schnell. Somit sind in der Robotertechnik flüssige Bewegungen möglich. In der KI-Forschung ist es schon lange das Ziel, eine preiswerte und lernfähige Kamera zu entwickeln. In der Vergangenheit waren für eine solche Technologie aber grosse Computer mit hochentwickelter Software notwendig. So sind die Kameras nur in Labors oder Forschungseinrichtungen mit grossem Budget eingesetzt worden. Die kleine und schnelle Pixy Cam2 ist problemlos mit einem Arduino oder Raspberry Pi zu verbinden. Die Kamera wird mit dem mitgelieferten Kabel über den ISP-Anschluss an das Arduino angeschlossen. Auch andere Anschlussarten sind mit der Kamera möglich. Wie beim Arduino gibt es für die Pixy Cam2 Softwarebibliotheken, die heruntergeladen werden können. So ist eine Verknüpfung schnell hergestellt und das Arduino kann die Daten der Kamera auswerten. Die Pixy Cam2 ist in der Lage, sieben verschiedene Objekte an der Form und Farbe zu erkennen und vergibt ihnen eine Signatur.

7.2.7 Evaluation Übermittlungsgerät



Abbildung 7: ipONEgprs

Beim Übermittlungsgerät wird mit dem ipONEgprs auf ein firmeninternes Produkt zurückgegriffen. Dieses Gerät wird in zahlreichen Brand- oder Einbruchsmeldeanlagen erfolgreich verbaut. Aus Datenschutzgründen kann das Gerät nur kurz und nicht umfassend beschrieben werden.

Tabelle 9: Evaluation Übermittlungsgerät

Typ	ipONEgprs
Verbindungsarten	
Verbindungswege	IP und gprs
Übermittlungsarten	
Mögliche Übermittlungen	• Alarmleitstellen • Private Leitstellen • SMS • E-Mail • Pager-Nachrichten • Voice-Nachrichten
Kriterien	
Eingänge	8
Elektrische Werte	
Spannungseingang	10 bis 36V DC
Allgemein	
Masse L x B	160mm x 100mm

7.2.8 Beschreibung und Vorteile vom ipONEgprs

Das hier gezeigte Übermittlungsgerät ipONEgprs übermittelt automatisch Alarmmeldungen, Betriebszustände und Störmeldungen. Das Gerät kann auf drei Arten betrieben werden: Es kann über IP oder gprs übermitteln oder optimalerweise über beide Verbindungen gleichzeitig, wobei die IP priorisiert wird und das gprs als Redundanz genutzt wird. Somit ist das Gerät redundanzfähig und kann bei einem Verbindungsunterbruch eines Kanals trotzdem noch an die Alarmzentrale übermitteln. Das Gerät kann SMS, E-Mail, Voice- oder Pager-Nachrichten senden. Mit seinen acht möglichen Kriterien ist es überall einsetzbar. Die Kriterien eins bis fünf sind frei wählbar und können mit Schleifenüberwachung oder als potenzialfreie Schleifen betrieben werden. Kriterien sechs bis acht sind ohne Schleifenüberwachung. Die Eingänge werden mit dem SysConf ipOne von der Firma TUS als Öffner oder Schliesser deklariert und programmiert.

7.2.9 Fazit Hardware

Nach gründlichem Vergleichen und Evaluieren der Hardware, kann die Passung auf die Anforderungen und die Komptabilität der Hardware-Komponenten im Gesamtsystem sichergestellt werden.

7.3 Pilotierung

Bei der Pilotierung geht es darum, die Hardware-Komponenten in ihrer Einzelfunktion und im Zusammenspiel zu testen. Hierfür werde ich das Arduino Uno mit der Pixy Cam2 verbinden und eine Testprojekt umsetzen. Nach der Evaluation des Testprojekts und den daraus resultierenden Anpassungen, geht es an die effektive Umsetzung der Arbeit.

7.3.1 Einzelfunktion

Zuerst muss die Software für das Arduino sowie für die Pixy Cam2 heruntergeladen werden. In meinem Fall die rot markierten Versionen für Windows.

Link: <https://www.arduino.cc/en/Main/Software>

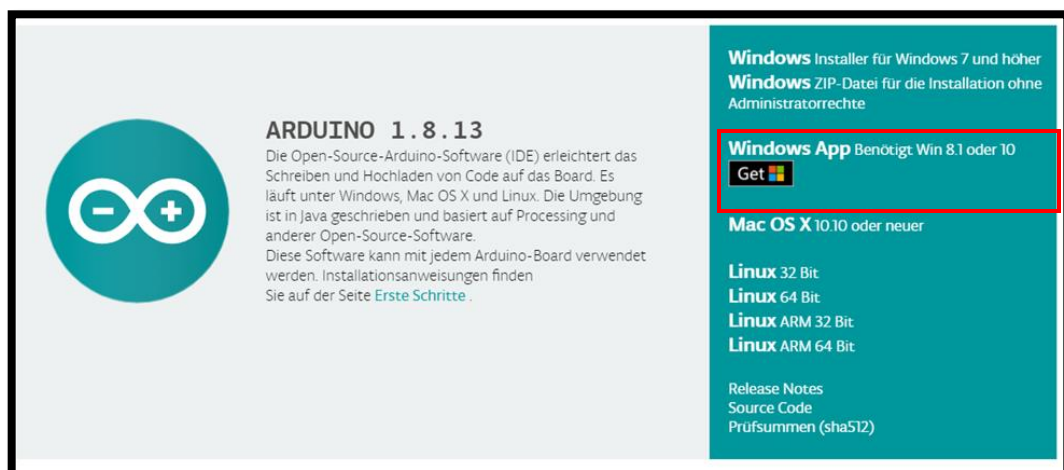


Abbildung 8: Software Arduino

Link: <https://pixycam.com/downloads-pixy2/>



Abbildung 9: Software Pixy Cam2

7.3.2 IDE Programmierumgebung für Arduino

Nachfolgend wird kurz das IDE-Tool erläutert. Die Programmierung vollzieht sich in drei unterschiedlichen Programmierabschnitten. In diesem beliebig ausgewählten Code zeige ich, wie das Programm aufgebaut ist.

Teil 1: Im Beispiel steht an dieser Stelle noch kein Inhalt. Hier werden später PINS aufgelistet, die im Programm immer wieder verwendet werden.

Teil 2: Im *void setup* wird der Befehl einmalig ausgeführt. Zum Beispiel ob der PIN ein Input oder Output ist.

Teil 3: Im *void loop* wird das Programmiererte in einer Dauerschleife immer wieder abgearbeitet und von Neuem gestartet.

Im *void loop* wird der Hauptteil der Programmierung stehen. Da es sich hier um eine Endlosschleife handelt, in welcher der Code abgearbeitet wird, empfehle ich hier den Code so kompakt wie möglich zu gestalten. Sollte der Code etwas lang werden, besteht die Möglichkeit, Bibliotheken aufzubauen. Damit der *loop* auf die Bibliothek springt, sollte eine Auswertung zutreffen. Mit diesem Vorgehen lässt sich der Code und der ganze Prozess kürzen und wird schneller abgearbeitet.



Abbildung 10: Arduino Programmierumfeld

7.3.3 Einbindung Pixy Cam2 in das Arduino-Tool

Für eine Verknüpfung der Pixy Cam2 mit dem Arduino-Tool wird eine Bibliothek bereitgestellt, welche man herunterladen und im Arduino IDE-Tool einpflegen kann. Das Installationsdokument stellt hierfür Erklärungen und Anleitungen bereit.

Link: <https://pixycam.com/downloads-pixy2/>

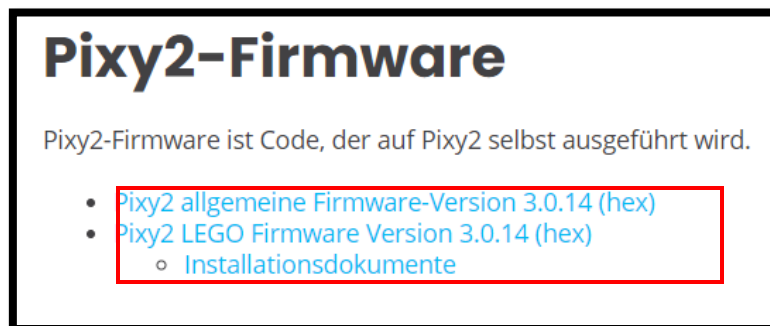


Abbildung 11: Pixy Cam2 Bibliothek

Nachdem man die Bibliothek im Arduino IDE-Tool eingebunden hat, kann man unter Datei -> Beispiele -> Pixy2 die Datei `ccc_hello_word` auswählen. Anschliessend wird der Code für die Pixy Cam2 ausgeführt.

```
#include <Pixy2.h>

// This is the main Pixy object
Pixy2 pixy;

void setup()
{
  Serial.begin(115200);
  Serial.print("Starting...\n");

  pixy.init();
}

void loop()
{
  int i;
  // grab blocks!
  pixy.ccc.getBlocks();

  // If there are detect blocks, print them!
  if (pixy.ccc.numBlocks)
  {
    Serial.print("Detected ");
    Serial.println(pixy.ccc.numBlocks);
    for (i=0; i<pixy.ccc.numBlocks; i++)
    {
      Serial.print("  block ");
      Serial.print(i);
      Serial.print(": ");
      pixy.ccc.blocks[i].print();
    }
  }
}
```

Abbildung 12: Code Pixy Cam2

7.3.4 Zusammenspiel

Nachdem nun die Software heruntergeladen und mit den Hardware-Komponenten zu einem Gesamtsystem verbunden wurde, sind die Vorbereitungen für das Testprojekts (Pilotierung) abgeschlossen. Die Umsetzung des Testprojekts hat zum Ziel, den Aufbau zu realisieren und dabei auszutesten, ob die Pixy Cam2 die erkannten Objekte an Arduino weiterleitet. Ausserdem soll erprobt werden, ob Arduino die erkannten Objekten korrekt mittels der LED anzeigt.

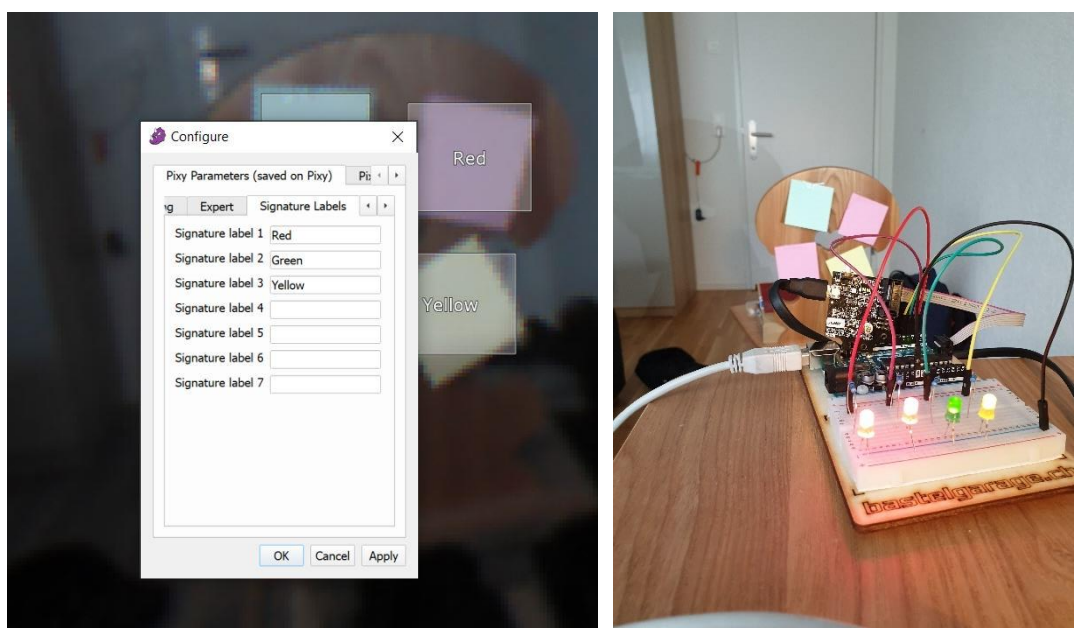


Abbildung 13: Testprojekt Arduino & Pixy Cam2

7.3.5 Fazit Pilotierung

Die Pilotierung hat gezeigt, dass die Hardware zusammen mit der Software optimal funktioniert. Die erstmalige Einbindung der Pixie Cam2-Bibliothek in das IDE-Tool hat sich als herausfordernd herausgestellt. Probleme konnten jedoch durch mehrmaliges Austesten behoben werden. Daraufhin hat das Zusammenspiel fehlerfrei und problemlos funktioniert. Dieser Testlauf umfasste erste Arbeitsschritte, welche im Projekt analog umgesetzt werden, die erlangten Erfahrungen im Umgang werden darin einfließen und zur Qualitätssicherung beitragen. Auf Basis dieser Vorbereitung kann nun die Umsetzung des Projekts erfolgen.

7.4 Vorstudie Umsetzung

Nach Abschluss der Pilotierung, wird nun die Vorstudie für die Projektumsetzung beschrieben. Hierfür wird zunächst ein Kontextdiagramm für das vorliegende Projekt erstellt.

7.4.1 Kontextdiagramm Vorstudie Umsetzung

Mit Hilfe eines Kontextdiagramms kann aufgezeigt werden, wie das System arbeiten soll.

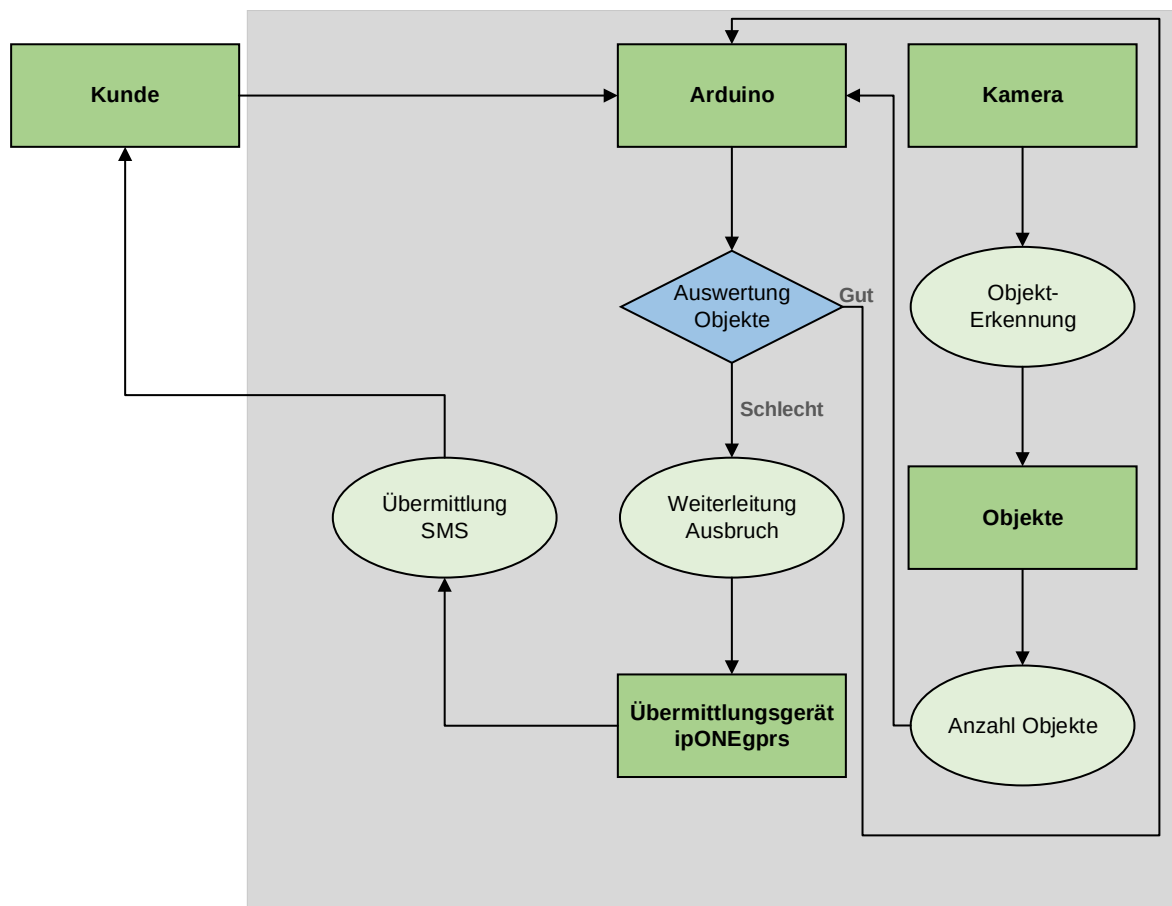


Abbildung 14: Kontextdiagramm Vorstudie Umsetzung

Die Kamera soll die Objekte erkennen und die Anzahl der Objekte an das Arduino weiterleiten. Das Arduino wertet die gelieferten Daten aus und entscheidet, ob die Auswertung gut oder schlecht ist. Beim Prädikat *gut* folgt eine Endschleife und beim Prädikat *schlecht* wird die Übermittlung ausgelöst. Der Kunde erhält in der Folge eine Benachrichtigung auf seinem Telefon.

7.4.2 Blockdiagramm Vorstudie Umsetzung

Das Blockdiagramm zeigt auf, wie die Hardware-Komponenten im Projekt miteinander verbunden sind.

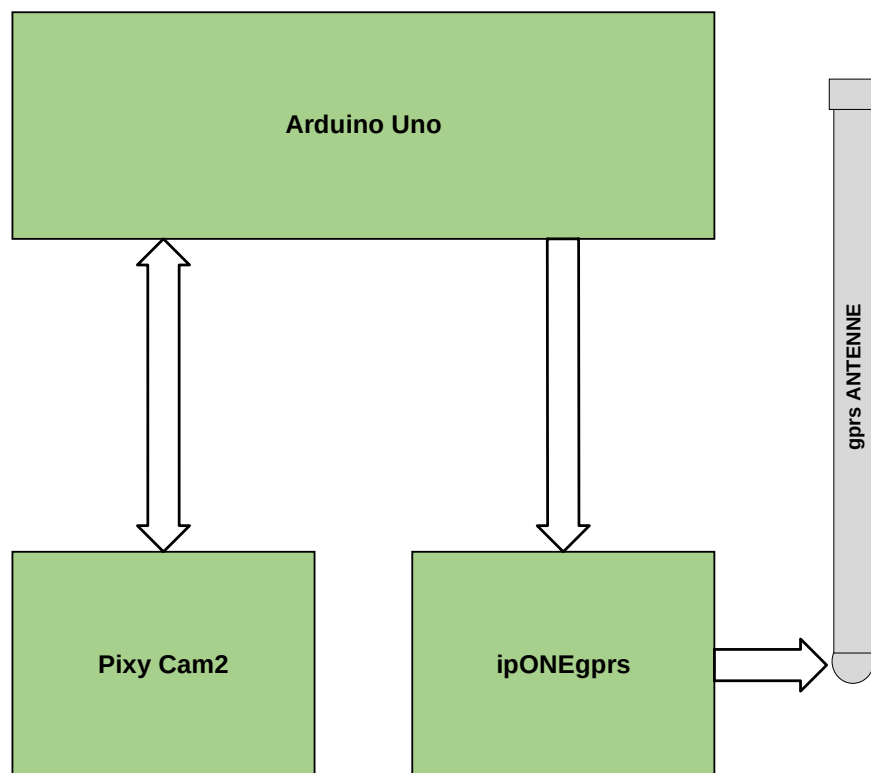


Abbildung 15: Blockdiagramm Vorstudie Umsetzung

7.4.3 Fazit Vorstudie Umsetzung

In der Phase der Vorstudie konnte der Ablauf konkretisiert werden. Die Kamera und der Mikrocontroller sind kompatibel und stellen kein Problem dar. Weiterer Klärungsbedarf besteht bei der Übermittlung des Ausbruchsignals auf das Übermittlungsgerät. Die Kontakte beim Übermittlungsgerät müssen potenzialfreie sein.

III Phase Detailkonzept

8 Aufbau Prototyp

Auf Basis der Erkenntnisse kann nun der Prototyp des Projekts aufgebaut und mit zusätzlichen Hardware-Komponenten ausgestattet werden. Am Ende des Prozesses resultiert ein funktionierendes System, dass beim Kunden montiert, in Betrieb genommen und getestet werden kann.

8.1 Erweiterung Hardware

Die Hauptkomponenten wurden in der Grobkonzept-Phase sorgfältig evaluiert. Nun benötigt das Gesamtsystem noch zusätzliche Hardware Komponente, namentlich ein Relais und einen Schalter. Diese werden nachfolgend beschrieben.

8.1.1 Erweiterung 4-Kanal Relais

Für das Übermittlungsgerät ipONEgprs wird zusätzlich ein Relais für das Alarmsignal benötigt, welches das Arduino übermittelt. Das Übermittlungsgerät kann nur potenzialfreie Kontakte auswerten, deshalb muss zwischen diesen Hardwarekomponenten ein Relais eingebaut werden. Nach meinen Recherchen habe ich mich für ein 4-Kanal Relais entschieden. Das Hauptargument für diesen Entscheid liegt in den drei Reserveplätzen, welche eine allfällige Systemerweiterung zu einem späteren Zeitpunkt erlauben.

Tabelle 10: Technische Daten 4-Kanal Relais

Technische Daten über das 4-Kanal Relais	
Spannungseingang	3.3V - 5V
Relais	4
Schaltleistung DC	30V / 10A
Schaltleistung AC	250V / 10A
Arduino kompatibel	Ja
Masse L x B	76mm x 56mm
Preis	CHF 12.00



Abbildung 16: 4-Kanal Relais

8.1.2 Erweiterung ON-OFF Schalter

Zum Blockieren des Übermittlungsgeräts wird einen Schalter benötigt. Der Schalter verfügt über drei Anschlusskontakte. So ist der Schalterzustand offen oder geschlossen. Mit einer Widerstandverschaltung kann das Arduino den Schalterzustand verarbeiten und ermitteln.



Abbildung 17: ON-OFF Schalter

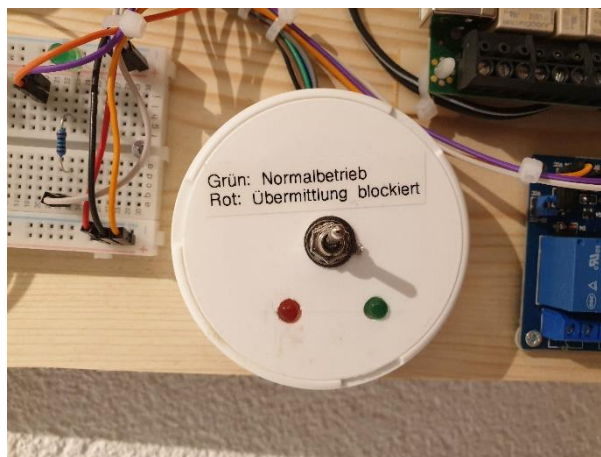


Abbildung 18: ON-OFF Schalter mit LEDs

Im vorliegenden Projekt ist der Schalter in einer Dose integriert worden und kann mit Hilfe der LEDs den Schalterzustand anzeigen.

8.1.3 Fazit Erweiterung

Da ich nun alle zusätzlich benötigte Hardware-Komponenten eruiert habe, und diese als unproblematisch im Zusammenspiel mit den übrigen Hardware-Komponenten angesehen werden können, kann nun der Aufbau des Prototypen beginnen. Das Relais wird für die Alarmweiterleitung und der Schalter für das Blockieren des Übermittlungsgeräts benötigt.

8.2 Einbindung Übermittlungsgerät

In diesem Kapitel binde ich das Übermittlungsgerät ipONEgprs in das System ein. Das Übermittlungsgerät wird mit dem Software-Tool SysConf der Firma TUS programmiert.

8.2.1 Übermittlungsgerät Tool SysConf ipONE

Als erstes lege ich die Kundendaten an. Über den Navigationsbaum auf der linken Seite kann in das entsprechende Register der Konfiguration bearbeiten.

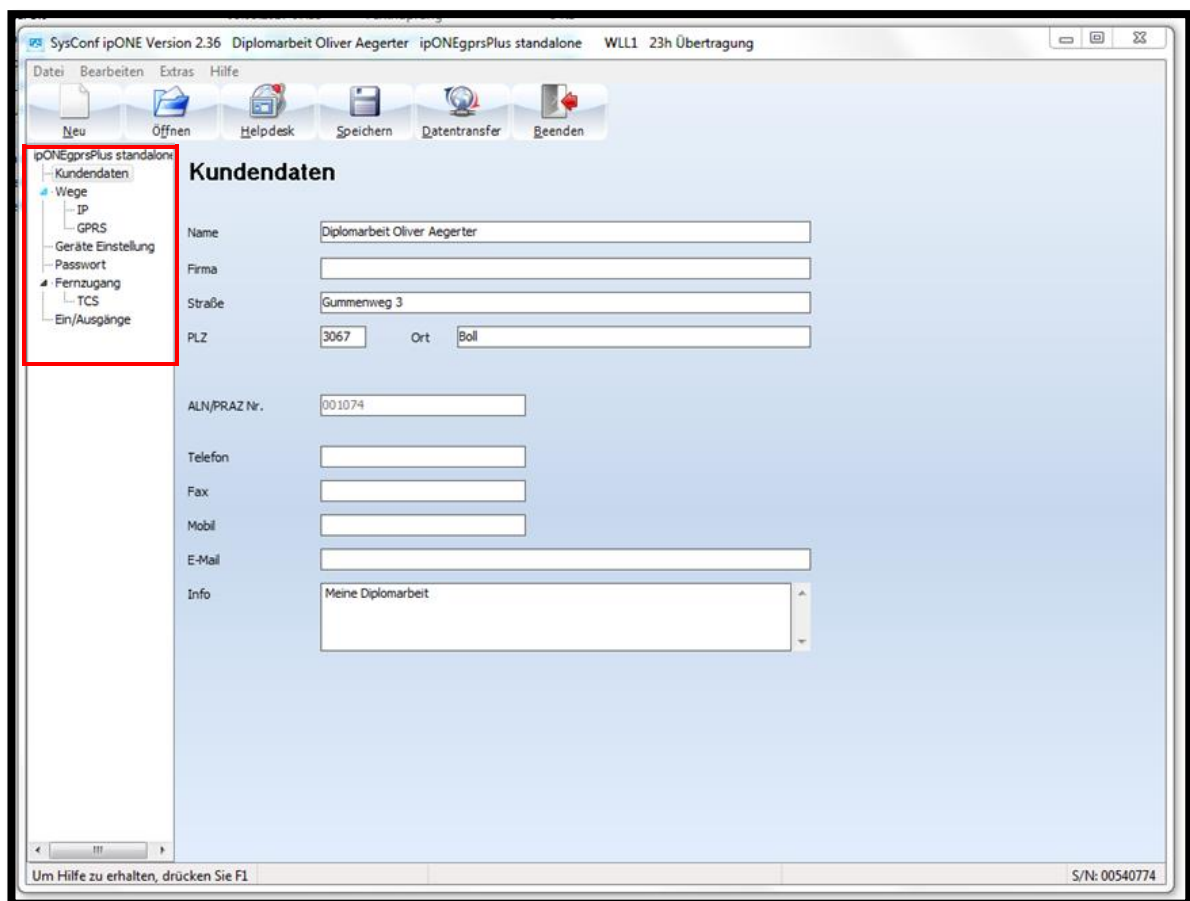


Abbildung 19: SysConf ipONE Kundendaten

Im Navigationsbaum unter *GPRS* muss die Funktion auf aktiv gestellt werden, nun gilt es die PIN der SIM-Karte zu hinterlegen. Hierbei gilt es zu bedenken, dass die SIM-Karte gesperrt wird, wenn die PIN dreimal falsch oder nicht eingegeben wird. Sollte dies passieren, muss man die Karte mithilfe eines Mobilgeräts wieder entsperrt werden.

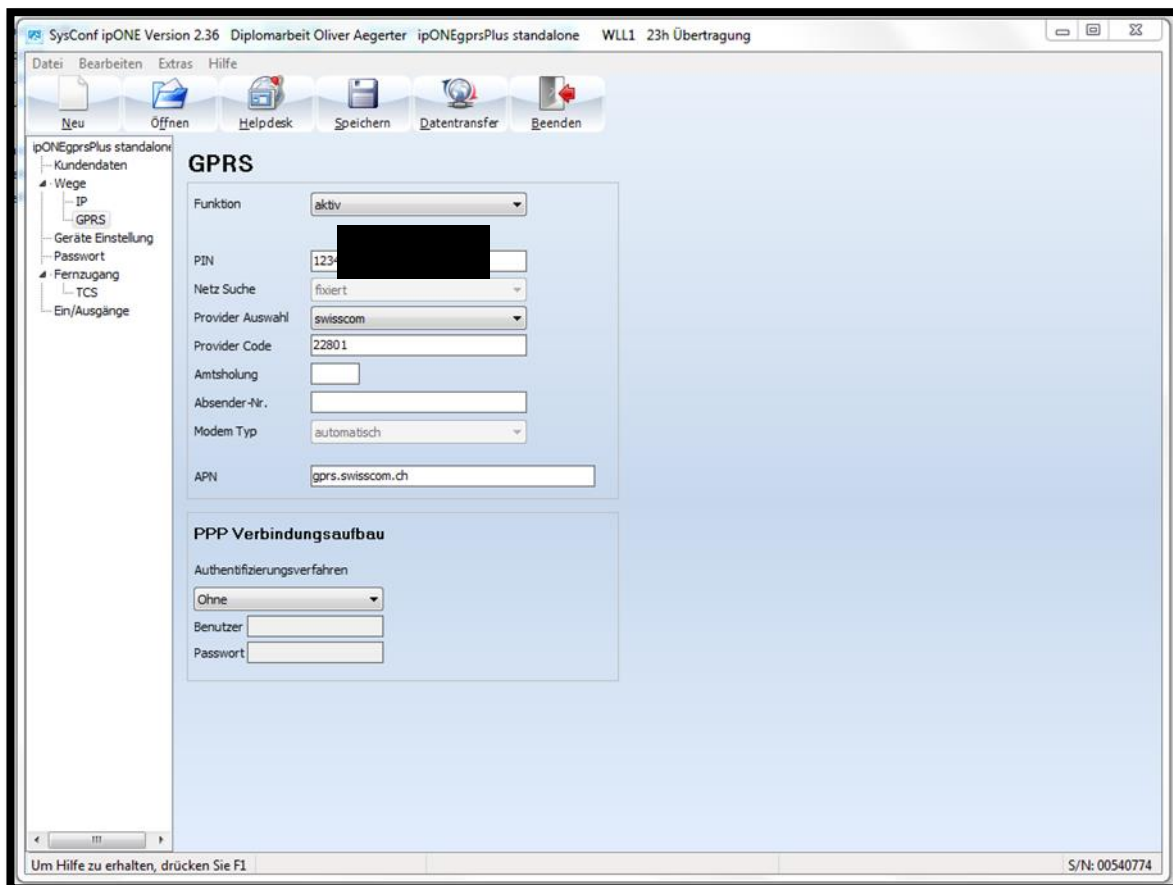


Abbildung 20: SysConf ipONE gprs

Im Navigationsbaum unter *Ein/ Ausgänge* wird nun auf dem *Eingang 1* das Alarmkriterium konfiguriert. Für Testzwecke gebe ich hier meine Daten ein und wechsle später die Daten auf die meines Kunden.

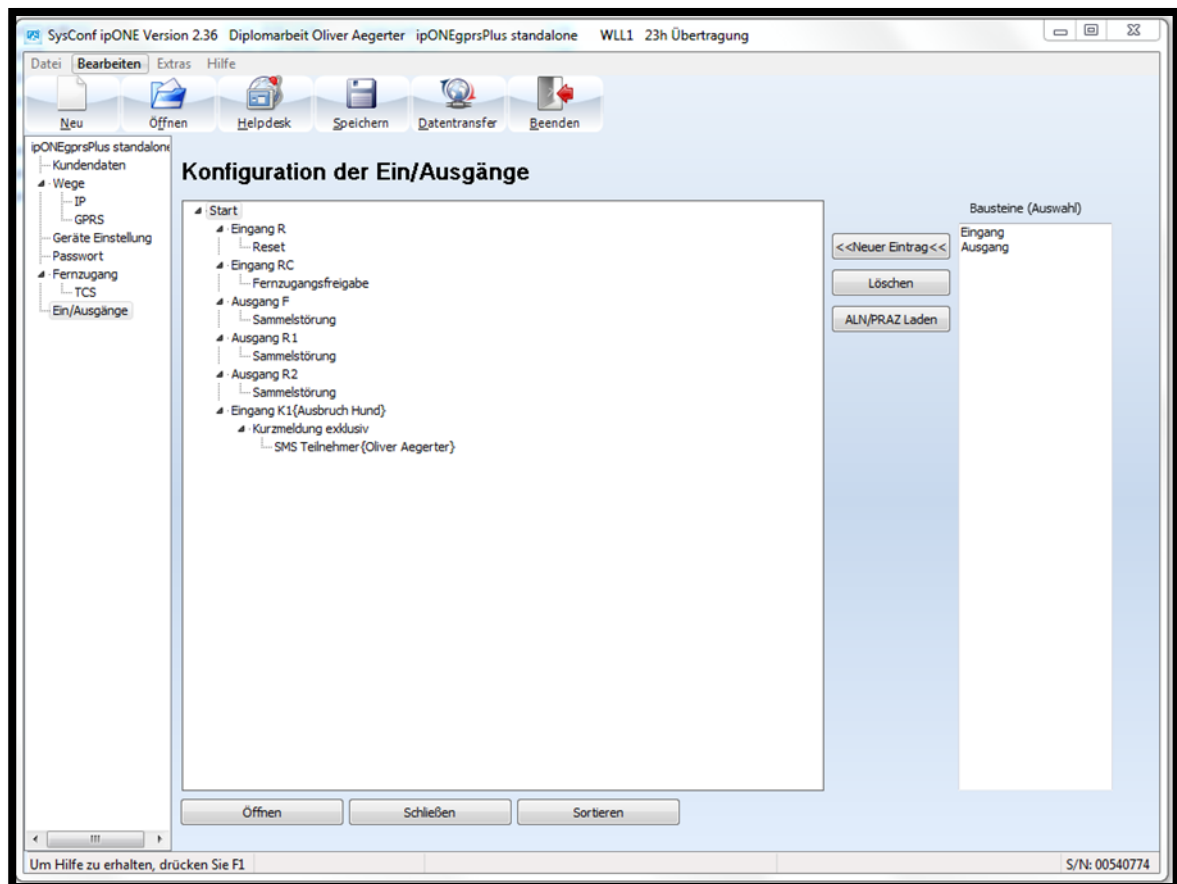


Abbildung 21. SysConf ipONE Ein/Ausgänge

In einem ersten Schritt wird nun der Teilnehmer hinzugefügt.

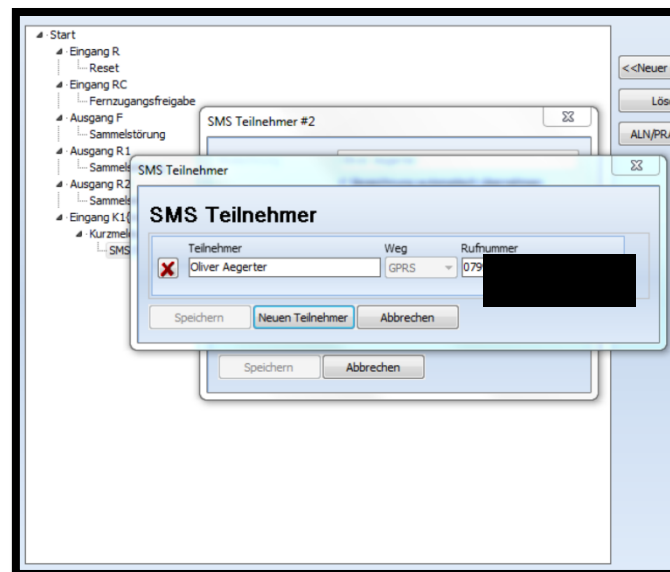


Abbildung 22: SysConf ipONE Teilnehmer

Unter den *Eingangskonfigurationen* können nun spezifischen Einstellungen vorgenommen werden. Für das System setze ich die Bezeichnung der abzusetzenden SMS ein, in diesem Fall «Ausbruch Hund». Ausserdem stelle ich eine Alarmweitergabeverzögerung von einer Minute ein.

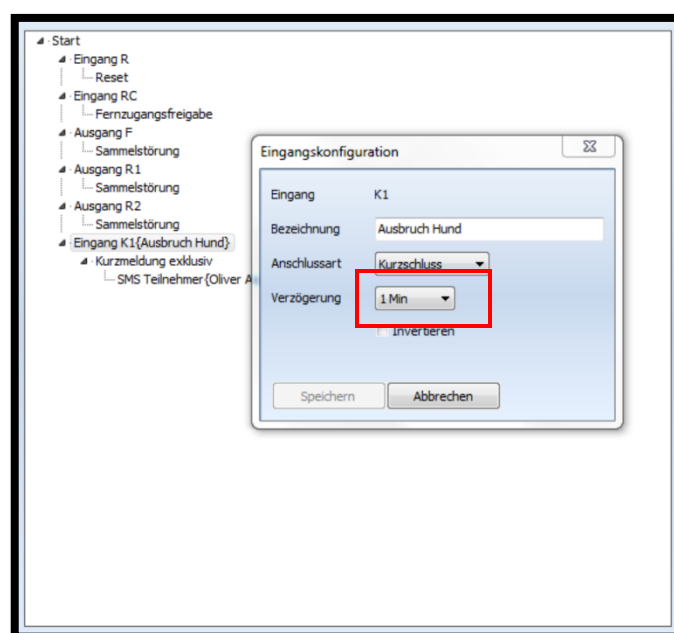


Abbildung 23: SysConf ipONE Alarmverzögerung

8.2.2 SysConf ipONE Helpdesk

Der Helpdesk ist ein nützliches Werkzeug, welches den Gerätezustand aufzeigt. Auf dem ersten Bild ist der Status des Eingangs 1 grün markiert, somit hat das Gerät keinen Alarm anstehend.

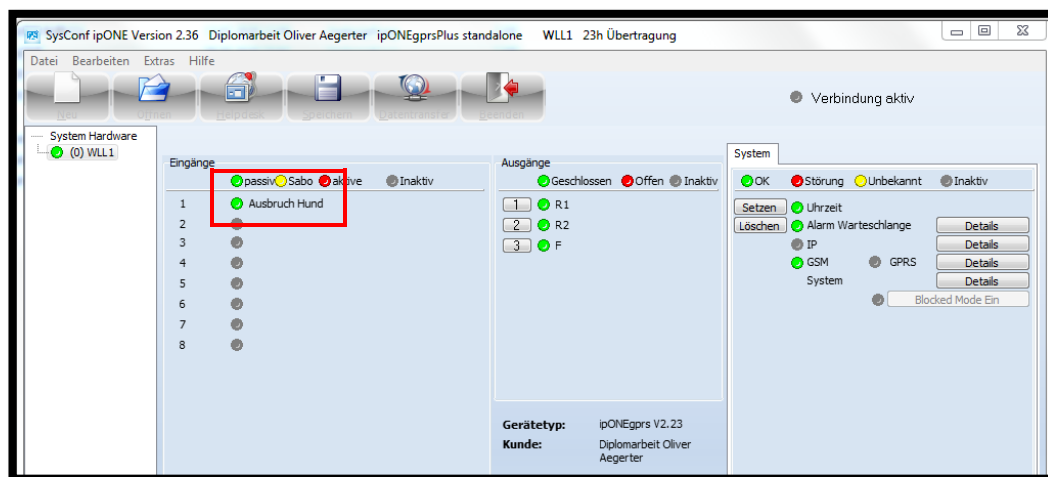


Abbildung 24: SysConf ipONE Helpdesk ohne Alarm

Nach dem Öffnen des Kontakts wechselt der Status auf Rot. Nun läuft die Verzögerung von einer Minute, nach Ablauf dieser Minute sendet mir das Gerät eine Alarm-SMS mit dem Text «Ausbruch Hund». Die Verzögerung wurde bewusst programmiert, falls ein Welpen aufgrund äußerer Umstände nicht detektiert werden kann, so zum Beispiel, wenn er unter der Mutter steht.

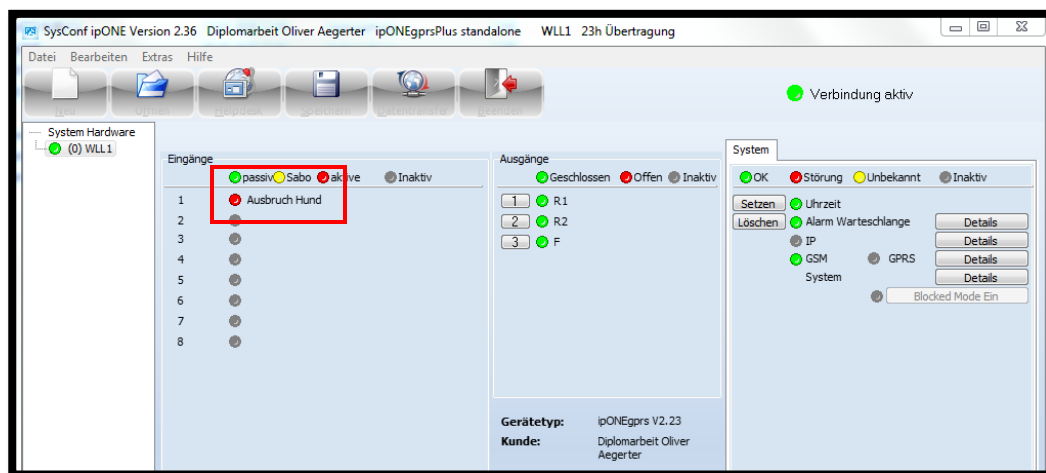


Abbildung 25: SysConf ipONE Helpdesk mit Alarm

8.2.3 Fazit Übermittlungsgerät

In meinem beruflichen Alltag beschäftige ich mich eingehend mit Übermittlungsgeräten. Insofern stellte die eben beschriebene Evaluation und der Aufbau des Übermittlungssystems keine besondere Herausforderung dar. Trotzdem galt es, das Problem der Übermittlung des Signals von Arduino Uno auf das Übermittlungsgerät zu lösen. Die effiziente Strategie, nämlich der Einbau des Relais, hat sich als sinnvoll herausgestellt. Folglich verlief der Testablauf des Übermittlungsgeräts erfolgreich und somit kann der nächste Projektschritt angegangen werden.

8.3 Aufbau und Umsetzung Prototyp

Nachdem die Hardware um die zusätzlichen Komponenten ergänzt worden ist, geht es darum, einen ersten Aufbau zu erstellen. Als Hilfestellung werden die definitiven Kontext- und Blockdiagramme erstellt.

8.3.1 Kontextdiagramm Prototyp

Das Kontextdiagramm zeigt die Alarmweitergabe sowie die programmierte Alarmweitergabe mit einer Minute Zeitverzögerung auf. Die Verzögerung dient der Vermeidung von Täuschungsalarmweitergaben. Es könnte beispielsweise sein, dass ein Welpen durch einen Gegenstand verdeckt wird und dadurch nicht mehr detektiert werden kann. Diese Verzögerung ist im Kapitel 8.2.1 mit der Programmierung des Übermittlungsgeräts realisiert worden.

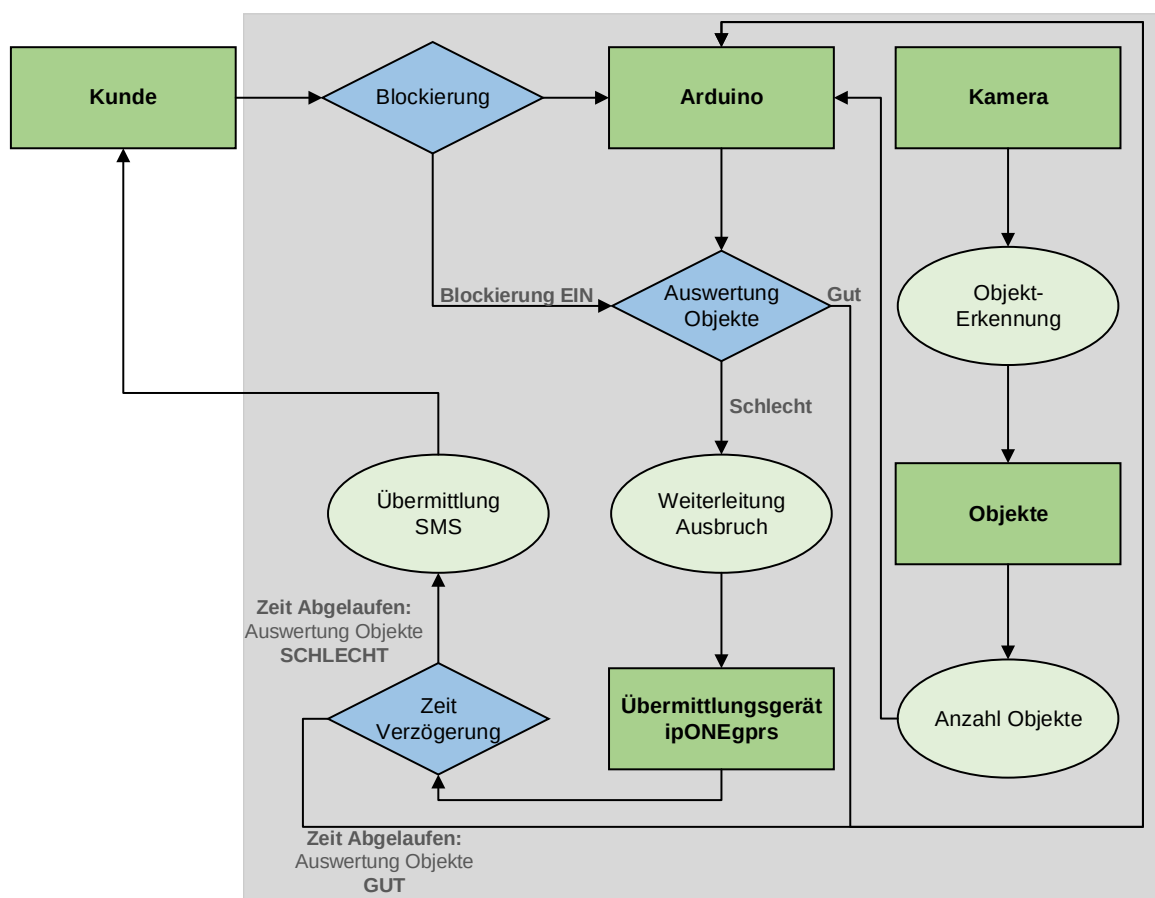


Abbildung 26: Kontextdiagramm Prototyp

8.3.2 Blockdiagramm Prototyp

Auf dem nachfolgenden definitiven Blockdiagramm sind nun auch die Schalter-Blockierung und das 4-Kanal Relais aufgeführt.

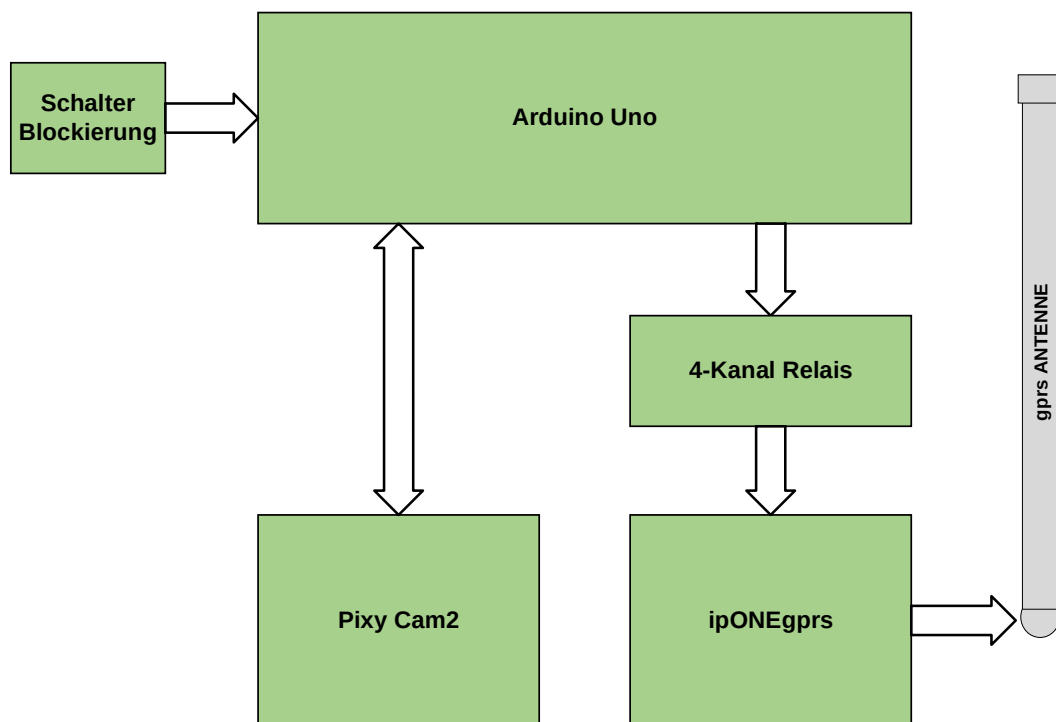


Abbildung 27: Blockdiagramm Prototyp

8.3.3 Der Aufbau

Zuhause verdrahte ich nun alle Hardware-Komponenten und kontrolliere eingehend ihre Funktionalität. Dieser Ablauf muss gründlich und korrekt ablaufen, denn als nächstes wird das System beim Kunden montiert und getestet.



Abbildung 28: Aufbau Prototyp

8.3.4 Programmierung und Software-Test

Nachdem ich den Prototypen erfolgreich aufgebaut habe, sende ich Arduino Uno die Programmierung. Im PixyMon-Tool kontrolliere ich, ob die Kamera Objekte erkennt. Dazu verwende ich Notizpapier in der Grösse 90mm x 90mm. Das Format ist bewusst gewählt, da die Kamera später aus höherer Distanz auch kleine Sachen erkennen muss. Der Abstand des gezeigten Bilds beträgt 1 Meter. Beim Kunden wird die Kamera auf 2 Meter montiert und in Betrieb genommen.

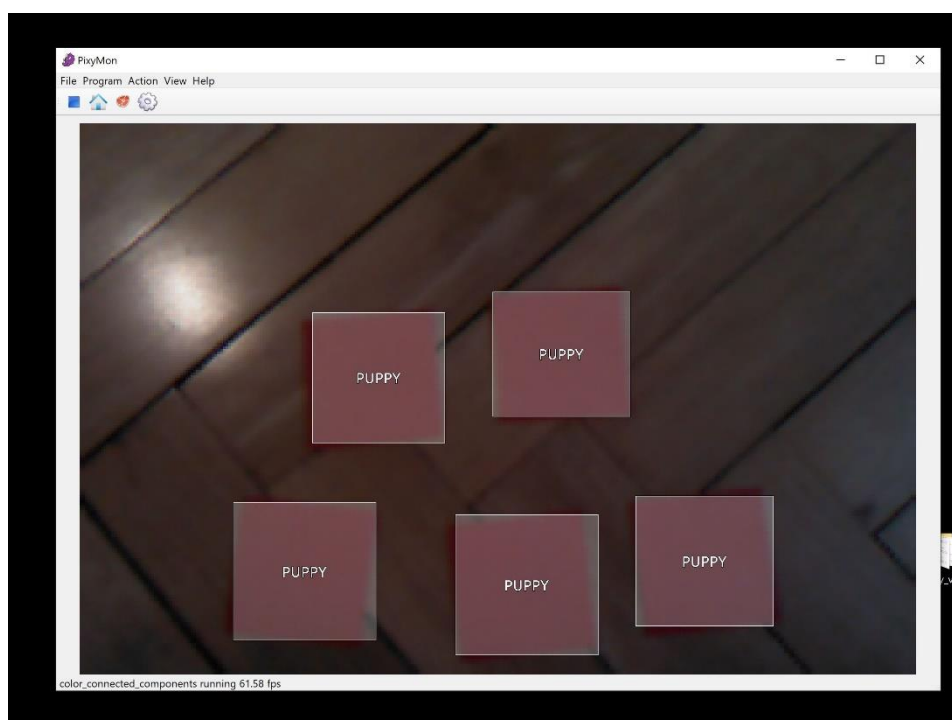


Abbildung 29: PixyMon Tool

8.3.5 Fazit Aufbau Prototyp

Nach dem Aufbau bin ich überzeugt, dass ich nicht mit grösseren Problemen beim Kunden rechnen muss. Das Gesamtsystem wurde gründlich vorbereitet und kann nun beim Kunden montiert werden.

8.4 Testkonzept

Im Testkonzept wird die Testung des Zusammenspiels und der Funktionalität der Hard- und Software konzipiert. Dabei werden die Testziele, der Testrahmen und das Testvorgehen, die Fehlerklassifizierung, die Testtabelle und der Testplan definiert.

8.4.1 Testziele

Im Vorfeld wurden *MUSS*- und *KANN*-Ziele definiert (siehe Kapitel 4.6.2 und Kapitel 4.6.3). Das primäre Ziel ist die Erfüllung aller mit den Kunden ausgehandelten *MUSS*-Ziele. Zusätzlich wurden zwei *KANN*-Ziele im Rahmen des Projekts umgesetzt: So ist das System überall flexibel einsetzbar und der Projektzustand wird mit Hilfe von LEDs angezeigt.

8.4.2 Testrahmen

Die Tests werden nach der Inbetriebnahme durchgeführt. Die Inbetriebnahme findet beim Kunden vor Ort statt, der im Vorfeld sicherstellt, dass die Tiere sich durch die Installation nicht gestresst fühlen. Die Testdurchführung und Protokollierung entfällt auf den Studenten.

8.4.3 Testvorgehen

Nach der Realisierungsphase werden mittels *Black-Box-Testmethode* die Testfälle umgesetzt. Sollte ein gravierender Fehler auftauchen, muss der Fehler vor Ort analysiert und behoben werden. Nach der Fehlerbehebung muss zwingend das Gesamtsystem erneut getestet werden.

8.4.4 Fehlerklassifizierung

Die Fehler werden wie folgt klassifiziert:

Tabelle 11: Fehlerklassifizierung

Fehlerklasse	Fehlerstufe	Beschreibung
Kritisch	3	Schwerwiegender Fehler Der Fehler muss sofort behoben werden
Mittel	2	Funktioneller Fehler Der Fehler muss dringend behoben werden
Gering	1	Unkritischer Fehler Der Fehler kann zu einem späteren Zeitpunkt behoben werden
Gut	0	Kein Fehler Der Testfall konnte erfolgreich durchgeführt werden

Dieser Abschnitt beschreibt den Black-Box-Test und wurde wortwörtlich der angegebenen Quelle entnommen: Der **Black-Box-Test** bezeichnet eine Methode des Softwaretests. Hierbei werden Tests anhand der Spezifikation/ Anforderung entwickelt. D.h. Tests werden ohne Kenntnisse über die innere Funktionsweise/ Implementierung des zu testenden Systems entwickelt. Das Programm wird als Black Box behandelt. Nur nach aussen sichtbares Verhalten fließt in den Test ein. Im Gegensatz dazu werden White-Box-Tests mit Blick auf den implementierten Algorithmus entwickelt.

Quelle: <https://de.wikipedia.org/wiki/Black-Box-Test>

8.4.5 Testtabelle

Für die anstehenden Tests erstelle ich eine Testtabelle, diese dient der Protokollierung der einzelnen Testungen und erlaubt eine spätere Prüfung und Bewertung der Ziele.

Tabelle 12: Testtabelle

ID/Bezeichnung	T-XX	Titel
Beschreibung	Beschreibung des Testfalls	
Testvoraussetzung	Text	
Durchführung	Beschreibung	
Durchläufe geplant	XX	
Erwartetes Ergebnis	Resultat	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Fehlerklasse	Fehlerstufe
Festgestellter Fehler	Beschreibung	

8.4.6 Testplan

In Testplan wird gezeigt, in welcher Zeitspanne getestet wurde und wann die Ziele abgeschlossen worden sind.

Tabelle 13: Testplan

Nummer	Aktivität	Verantwortlich	Termin
1	Vorbereitung	Oliver Aegerter	07.10.2020
2	Testing	Oliver Aegerter	07.-08.10.2020
3	Abschluss	Oliver Aegerter	08.10.2020

8.4.7 Fazit Testkonzept

Das Testkonzept ist ein wichtiger Bestandteil eines Projekts. Es erleichtert die Vorgänge bezüglich der Tests sowie die anschliessende Dokumentation.

8.5 Kosten

Mit dem Auftragsgeber wurde kein definitives Limit besprochen. Die Kosten sollen sich im Rahmen von 300.00 CHF bewegen.

Tabelle 14: Kosten

Artikel	Lieferant	Anzahl	Preis pro Stück in CHF	Kosten in CHF
Pixy Cam2	Bastelgarage.ch	1	77.50	77.50
Arduino Uno	Bastelgarage.ch	1	27.90	27.90
Zubehör für Arduino	Bastelgarage.ch	2	15.00	30.00
4-Kanal Relais Modul	Bastelgarage.ch	1	12.00	12.00
Teleskopstange 80cm	OBI	1	7.95	7.95
Rohrschellen	OBI	2	3.90	7.80
Kleinmaterial	OBI	1	2.10	2.10
Brett 50x50	Bau&Hobby	1	28.75	28.75
USB Verlängerung	Digitec	2	9.80	19.60
Universal- Netzteil	Digitec	1	18.00	18.00
ipONEgprs	Securiton AG	1	0	0
Total				231.60

Mit den Materialkosten von 231.60 CHF bewege ich mich im angestrebten Kostenrahmen für das Projekt. Doch konnte das Ziel nur erreicht werden, weil ich das Übermittlungsgerät gratis über meine Firma beziehen konnte. Ansonsten wären zusätzliche Kosten von 1060.00 CHF aufgelaufen und damit das Limit deutlich überschritten.

8.5.1 Fazit Kosten

Die aufgelaufen Kosten befinden sich im Rahmen der veranschlagten 300.00 CHF. Dieser tiefe Preis konnte dank den vorteilhaften Konditionen der Arduino-Hardware-Komponenten und selbstredend durch die Zurverfügungstellung des ipONEgprs erzielt werden.

III Phase Realisierung

9 Realisierung beim Kunden

Nachdem nun alle Vorbereitungsarbeiten abgeschlossen sind, wird das System beim Kunden montiert und getestet. In diesem Kapitel geht es darum, den Aufbau und das Testen aufzuzeigen.

9.1 Aufbau beim Kunden

Das Gesamtsystem wird auf einer Höhe von rund zwei Metern montiert und für die Inbetriebnahme vorbereitet. Nachdem die Installation abgeschlossen worden ist, wird das System eingehend geprüft. Nach Abschluss der Tests wird das System dem Kunden übergeben, danach soll das Gesamtsystem autonom laufen und keinerlei Störungen aufweisen. Ich stehe dem Kunden jedoch bei Problemen zur Verfügung.



Abbildung 30: Aufbau beim Kunden

9.1.1 Serielle Schnittstelle im IDE-Tool

Im Arduino IDE Tool unter *Werkzeuge* sehen wir im seriellen Monitor was gerade ausgewertet wird. Vertikal werden die detektierten Welpen angezeigt und horizontal der Schalterzustand für die Blockierung.



Abbildung 31: Serielle Schnittstelle im IDE-Tool

9.1.2 Das Projekt

Hier erläutere ich kurz den Aufbau meines Projekts. Es besteht aus einer Ober- und Unterseite. Das Gesamtsystem wurde auf einem Holzbrett mit den Maßen 50cm x 50cm montiert.

Die Unterseite:

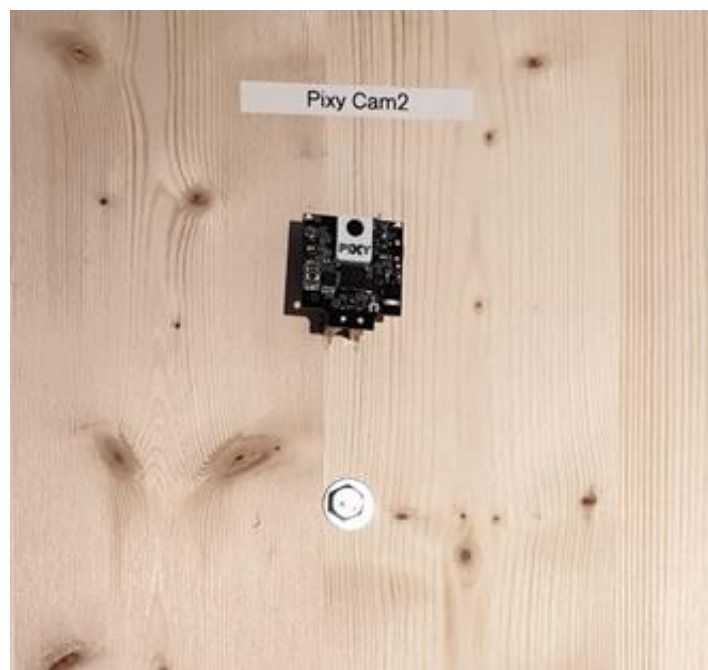


Abbildung 32: Aufbau Pixy Cam2

Auf der Unterseite ist lediglich die Pixy Cam2 montiert. Sie detektiert die Objekte und sendet an Arduino weiter.

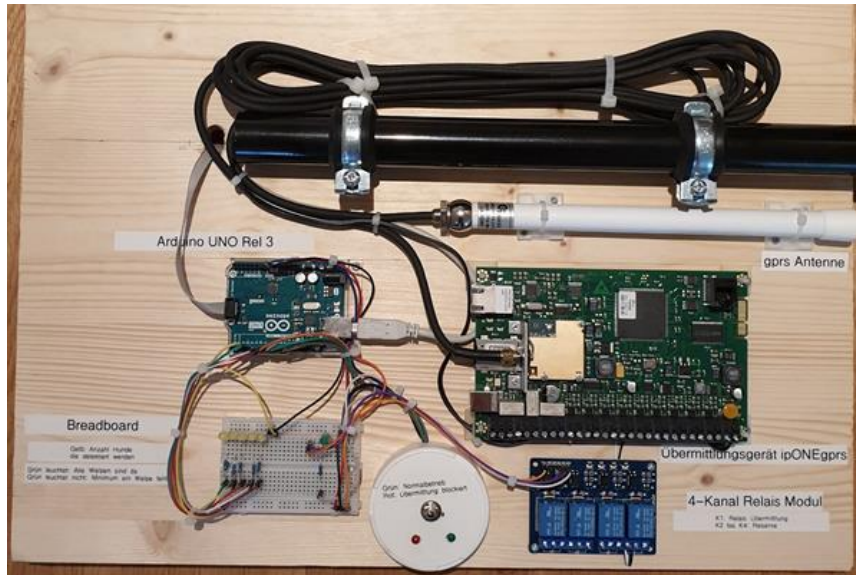
Die Oberseite:

Abbildung 33: Aufbau Hardware-Komponente

Auf der Oberseite befindet sich das restliche System, dieses wird nun kurz erläutert:

- **Arduino Uno Rel 3**
 - Das Herzstück des Systems. Auf dem Arduino ist die Software programmiert, es schaltet und verwaltet das System.
- **Breadboard**
 - Auf dem Breadboard sind 5 gelbe LEDs verbaut. Jedes LED steht für einen Welpen. Das Grüne LED zeigt nur an, ob alle fünf Welpen detektiert werden.
- **Schalter**
 - Der Blockier-Schalter ist mit einem roten und grünen LED ausgestattet. Die LEDs zeigen den Schalterzustand an.
- **4-Kanal Relais Modul**
 - Übermittelt das Signal vom Arduino auf das Übermittlungsgerät. Drei Kanäle sind als Reserve für Erweiterungen gedacht.
- **Übermittlungsgerät ipONEgprs**
 - Das Übermittlungsgerät sendet bei einem Ausbruch eines oder mehrerer Welpen dem Kunden eine Alarm-SMS
- **gprs Antenne**
 - Übermittelt die Nachricht (SMS) an den Kunden.

9.1.3 Stromlaufdiagramm des Projekts

Das Stromlaufdiagramm zeigt auf, wie das Projekt verdrahtet worden ist.

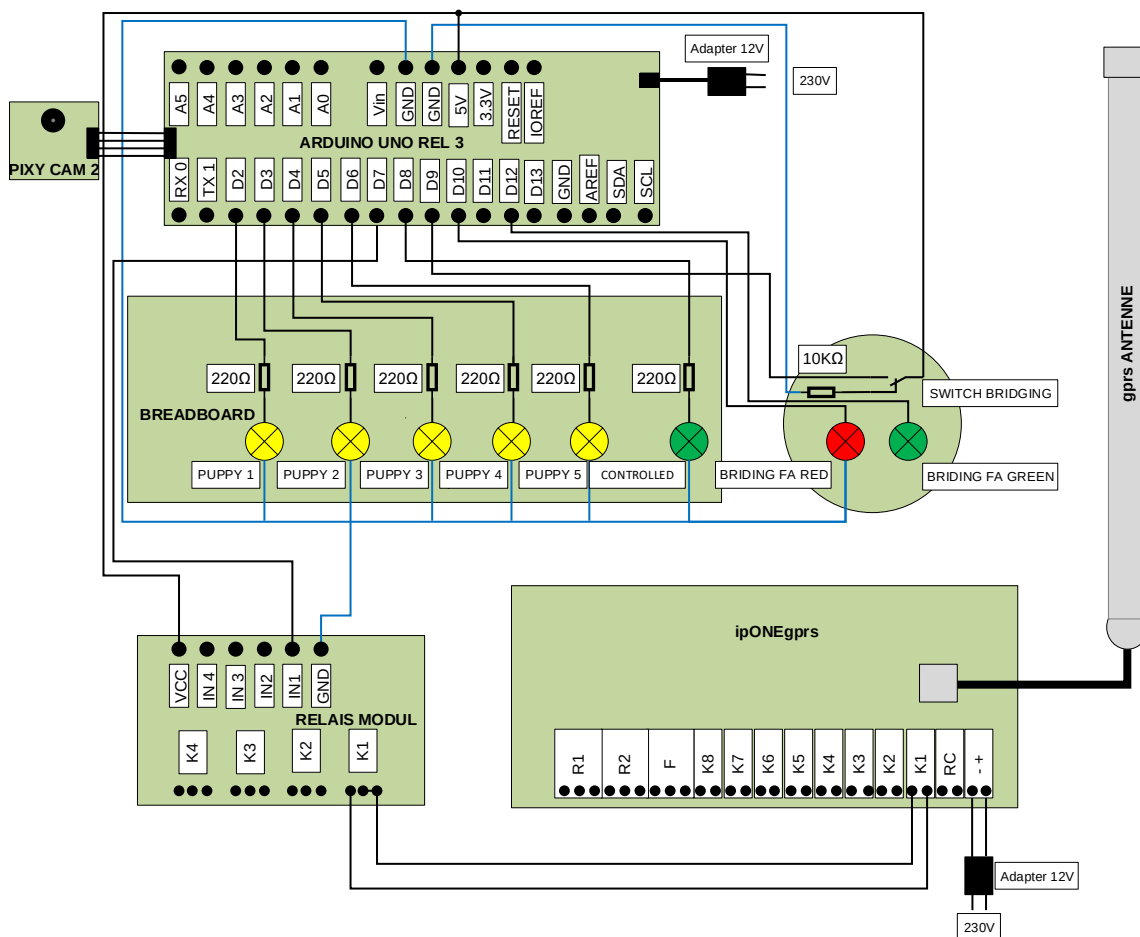


Abbildung 34: Stromlaufschema des Projekts

9.1.4 Fazit Aufbau beim Kunden

Der Aufbau meines Systems hat nahezu reibungslos funktioniert. Jedoch war die Montagehöhe etwas hoch und könnte optimiert werden. Ausserdem hat sich herausgestellt, dass die Pixy Cam2 mit dem Verlängerungskabel nicht mehr funktioniert, insofern musste die Oberflächengröße des Bretts so gewählt werden, dass darauf ebenfalls der Laptop Platz finden konnte.

9.2 Der Code

In diesem Kapitel zeige ich ein Flussdiagramm und den programmierten Code. Der Code wird zusätzlich erläutert.

9.2.1 Flussdiagramm Software

Das Flussdiagramm zeigt welche Aufgaben die Software erfüllt und abarbeitet.

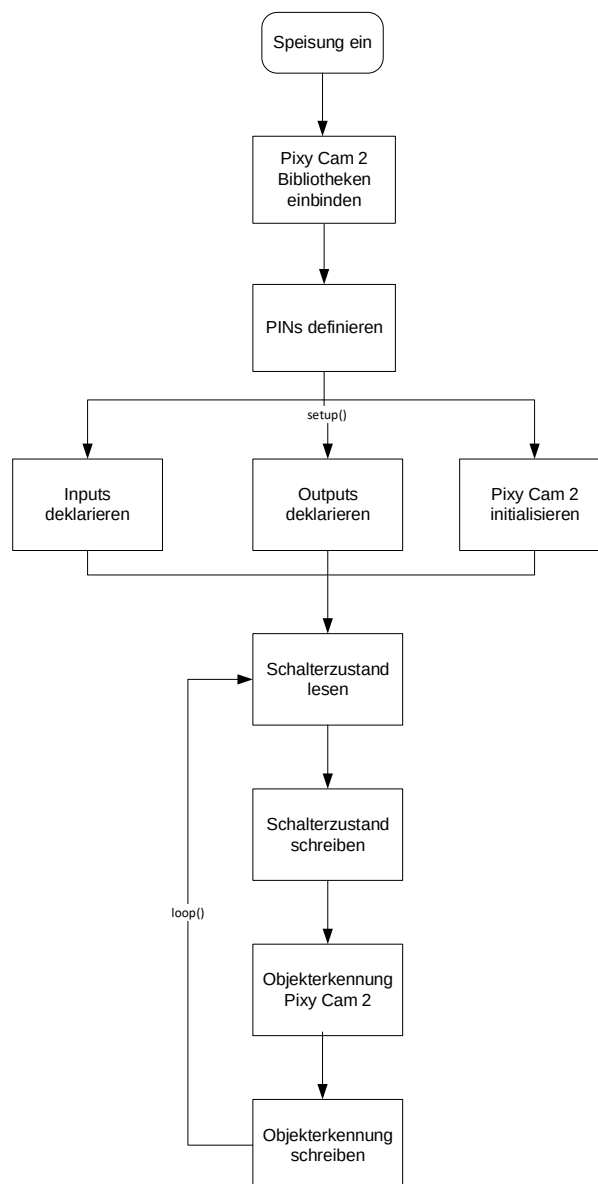


Abbildung 35: Flussdiagramm Software

9.2.2 Der programmierte Code

Wie Im Kapitel 7.3.2 schon erläutert, besteht der Code aus drei Teilen. Nachfolgend wird der Code für die Welpenüberwachung erläutert.

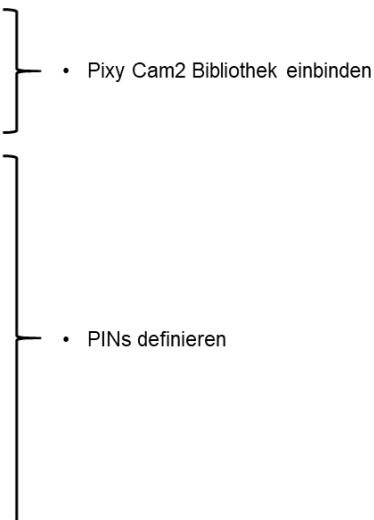
Teil 1:

Hier werden Befehle aufgelistet, die im Programm immer wieder verwendet werden.

```
#include <Pixy2.h>

Pixy2 pixy;

int PUPPY_LED1 = 2;
int PUPPY_LED2 = 3;
int PUPPY_LED3 = 4;
int PUPPY_LED4 = 5;
int PUPPY_LED5 = 6;
int TRANSMISSION = 7;
int CONTROLLED_GREEN = 8;
int SWITCH_STATE;
int InputPin = 9;
int BRIDING_FA_RED = 10;
int BRIDING_FA_GREEN = 11;
int numberOfPUPPY = 0;
```



The diagram shows two vertical curly brackets on the right side of the code. The first bracket spans the first two lines of code and is associated with the bullet point 'Pixy Cam2 Bibliothek einbinden'. The second bracket spans the remaining ten lines of code and is associated with the bullet point 'PINs definieren'.

- Pixy Cam2 Bibliothek einbinden
- PINs definieren

Abbildung 36: Der Code Teil 1

Teil 2:

Im *void setup* wird der Befehl einmalig ausgeführt. Zum Beispiel ob der PIN ein Input oder Output ist.

```
void setup()
{
    pinMode(InputPin, INPUT);

    |
    pinMode(PUPPY_LED1, OUTPUT);
    pinMode(PUPPY_LED2, OUTPUT);
    pinMode(PUPPY_LED3, OUTPUT);
    pinMode(PUPPY_LED4, OUTPUT);
    pinMode(PUPPY_LED5, OUTPUT);
    pinMode(CONTROLLED_GREEN, OUTPUT);
    pinMode(TRANSMISSION, OUTPUT);
    pinMode(BRIDING_FA_RED, OUTPUT);
    pinMode(BRIDING_FA_GREEN, OUTPUT);

    Serial.begin(115200);
    Serial.print("Starting...\n");

    pixy.init();
}
```

- } • Input PIN deklarieren
- } • Output PINs deklarieren
- } • Kommunikation starten
• Pixy Cam2 initialisieren

Abbildung 37: Der Code Teil 2

Teil 3:

Im *void loop* wird das Programmiererte in einer Dauerschleife immer wieder abgearbeitet und von Neuem gestartet. Ich habe bei der vorliegenden Codierung mit Bibliotheken gearbeitet.

```
void loop()
{
    Schalterzustand_schreiben();

    if (SWITCH_STATE == 1)
    {
        digitalWrite(BRIDING_FA_RED, HIGH);
        digitalWrite(BRIDING_FA_GREEN, LOW);
        digitalWrite(TRANSMISSION, LOW);

    }
    else
    {
        digitalWrite(BRIDING_FA_RED, LOW);
        digitalWrite(BRIDING_FA_GREEN, HIGH);

        int i;
    }
}
```

- } • Lesen und schreiben vom Schalterzustand
- } • Übermittlungs-LED Blockierung ein
• Übermittlungs-LED Betrieb aus
• Übermittlung blockiert
- } • Übermittlungs-LED Blockierung aus
• Übermittlungs-LED Betrieb aus

Abbildung 38: Der Code Teil 3



Abbildung 39: Der Code Teil 3.1

Bis hier ist der aktive Teil des *loop*. Nun springt der Code auf die Bibliotheken. Mit den Bibliotheken bewirke ich, dass der abzuarbeitende Code nicht zu lange ausfällt und so der *loop*-Prozess kürzer wird.

```

void Schalterzustand_schreiben()
{
    SWITCH_STATE = digitalRead(InputPin);
    Serial.print(" Schalterzustand ");
    if(SWITCH_STATE == 1)
    {
        Serial.print(" System aus ");
    }else
    {
        Serial.print(" System ein ");
    }
}

void turnOffAllLEDS()
{
    digitalWrite(PUPPY_LED1, LOW);
    digitalWrite(PUPPY_LED2, LOW);
    digitalWrite(PUPPY_LED3, LOW);
    digitalWrite(PUPPY_LED4, LOW);
    digitalWrite(PUPPY_LED5, LOW);
}

void turnOnOnePUPPYLED()
{
    digitalWrite(PUPPY_LED1, HIGH);
    digitalWrite(PUPPY_LED2, LOW);
    digitalWrite(PUPPY_LED3, LOW);
    digitalWrite(PUPPY_LED4, LOW);
    digitalWrite(PUPPY_LED5, LOW);
    delay(10);

    digitalWrite(CONTROLLED_GREEN, LOW);
    delay(3000);
    digitalWrite(TRANSMISSION, HIGH);
}

void turnOnTwoPUPPYLED()
{
    digitalWrite(PUPPY_LED1, HIGH);
    digitalWrite(PUPPY_LED2, HIGH);
    digitalWrite(PUPPY_LED3, LOW);
    digitalWrite(PUPPY_LED4, LOW);
    digitalWrite(PUPPY_LED5, LOW);
    delay(10);

    digitalWrite(CONTROLLED_GREEN, LOW);
    delay(3000);
    digitalWrite(TRANSMISSION, HIGH);
}

void turnOnTreePUPPYLED()
{
    digitalWrite(PUPPY_LED1, HIGH);
    digitalWrite(PUPPY_LED2, HIGH);
    digitalWrite(PUPPY_LED3, HIGH);
    digitalWrite(PUPPY_LED4, LOW);
    digitalWrite(PUPPY_LED5, LOW);
    delay(10);

    digitalWrite(CONTROLLED_GREEN, LOW);
    delay(3000);
    digitalWrite(TRANSMISSION, HIGH);
}

```

• Schalterzustand schreiben

• Alle LEDs aus

• Ein LED ein
• Vier LEDs aus

• Kontroll-LED aus
• Übermittlung ein

• Zwei LEDs ein
• Drei LEDs aus

• Kontroll-LED aus
• Übermittlung ein

• Drei LEDs ein
• Zwei Leds aus

• Kontroll-LED aus
• Übermittlung ein

Abbildung 40: Der Code Teil 3.2


```
void turnOnFourPUPPYLED()
{
    digitalWrite(PUPPY_LED1, HIGH);
    digitalWrite(PUPPY_LED2, HIGH);
    digitalWrite(PUPPY_LED3, HIGH);
    digitalWrite(PUPPY_LED4, HIGH);
    digitalWrite(PUPPY_LED5, LOW);
    delay(10);

    digitalWrite(CONTROLLED_GREEN, LOW);
    delay(3000);
    digitalWrite(TRANSMISSION, HIGH);
}

void turnOnFivePUPPYLED()
{
    digitalWrite(PUPPY_LED1, HIGH);
    digitalWrite(PUPPY_LED2, HIGH);
    digitalWrite(PUPPY_LED3, HIGH);
    digitalWrite(PUPPY_LED4, HIGH);
    digitalWrite(PUPPY_LED5, HIGH);
    delay(10);

    digitalWrite(CONTROLLED_GREEN, HIGH);
    delay(3000);
    digitalWrite(TRANSMISSION, LOW);
}
```

- Vier LEDs ein
- Ein LED aus
- Kontroll-LED aus
- Übermittlung ein
- Alle LEDs ein
- Kontroll-LED ein
- Übermittlung aus

Abbildung 41: Der Code Teil 3.3

9.2.3 Fazit Code

Ich habe mir die Programmiersprache C selbst beigebracht und somit war der Zeitaufwand grösser als erwartet. Schlussendlich entstand ein funktionierender Code.

9.3 Prüfung

In diesem Kapitel geht es um die eingehende Prüfung und Auswertung des Systems. Im Kapitel 4.6.2 und Kapitel 4.6.3 wurden *MUSS*- und *KANN*-Ziele aufgezeigt. Die *MUSS*-Ziele müssen zwingen erfüllt werden. Bei den *KANN*-Zielen werden nur diejenigen Ziele bewertet, die im Projekt umgesetzt worden sind. Sollten nach den Testlauf Fehler vorliegen, müssen diese analysiert werden und nach der Fehlerbehebung muss der Test wiederholt werden.

9.3.1 Testprotokolle 1 MUSS-Ziele

Es wurden im Projekt vier *MUSS*-Ziele definiert, diese werden nun geprüft.

Tabelle 15: Testprotokoll T-01

ID/Bezeichnung	T-01	Dauerhafte Überwachung der Welpen
Beschreibung	Das System muss die Welpen dauerhaft erkennen und überwachen.	
Testvoraussetzung	Das System ist in Betrieb und die Pixy Cam2 ist mit dem Arduino verbunden.	
Durchführung	Es wird mit dem PixyMon v2-Tool eine Verbindung mit dem Laptop hergestellt und kontrolliert, ob alle Objekte detektiert werden.	
Durchläufe geplant	5	
Erwartetes Ergebnis	Probleme mit der Detektieren der Objekte.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Mittel	2
Festgestellter Fehler	Kamera verliert manchmal die Objekte.	

Tabelle 16: Testprotokoll T-02

ID/Bezeichnung	T-02	Alarmierung erfolgt über ein Übermittlungsgerät
Beschreibung	Bei einem Ausbruch eines oder mehrerer Welpen wird der Kunde mit einer SMS benachrichtigt.	
Testvoraussetzung	Das Übermittlungsgerät ist im System integriert und hat Spannung.	
Durchführung	Es werden Objekte entfernt, folglich sollte eine Alarmierung über das Übermittlungsgerät stattfinden.	
Durchläufe geplant	3	
Erwartetes Ergebnis	Eine Alarmierung wird ausgelöst.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gut	0
Festgestellter Fehler	Keine	

Tabelle 17: Testprotokoll T-03

ID/Bezeichnung	T-03	Fehlerfreier Betrieb über die gewünschte Überwachungszeit der Welpen
Beschreibung	Das System muss fehlerfrei funktionieren.	
Testvoraussetzung	Das Gesamtsystem ist in Betrieb und das System läuft stabil und ohne Fehler.	
Durchführung	Das System ist autonom.	
Durchläufe geplant	5	
Erwartetes Ergebnis	Da T-01 nicht fehlerfrei funktioniert, erwarte ich hier einen Folgefehler, der sich mit der Behebung von T-01 automatisch auflöst.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gering	1
Festgestellter Fehler	Die Kamera verliert manchmal die Objekte und so werden Täuschungsalarme abgesetzt.	

Tabelle 18: Testprotokoll T-04

ID/Bezeichnung	T-04	System muss an- und abschaltbar sein
Beschreibung	Die Welpen sind nicht immer in der Wurfbox. Die Alarmierung muss daher blockierbar sein.	
Testvoraussetzung	Der Schalter ist im System integriert und die Schalterfunktion funktioniert	
Durchführung	Der Schalter wird in den Blockiermodus gestellt und es werden Objekte aus der Wurfbox genommen. Das System sollte nun keine SMS an den Kunden versenden.	
Durchläufe geplant	10	
Erwartetes Ergebnis	Einwandfreies Ein- und Ausschalten (Blockieren) vom Übermittlungsgerät.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gut	1
Festgestellter Fehler	Keine	

9.3.2 Testprotokolle 1 KANN-Ziele

Es wurden vier *KANN*-Ziele definiert, davon wurden zwei im Projekt umgesetzt. Die Resultate des Tests werden nachfolgend aufgezeigt.

Tabelle 19: Testprotokoll T-05

ID/Bezeichnung	T-05	Mit Hilfe von LEDs kann der Zustand des Projekt erkannt werden
Beschreibung	Die LEDs zeigen an, wie viele Welpen detektiert werden und ob der Schalter auf <i>ein</i> oder <i>blockiert</i> ist.	
Testvoraussetzung	Das System ist in Betrieb und Objekte werden detektiert.	
Durchführung	Schalter wird hin und her gestellt und Objekte werden aus der Wurfbox entnommen.	
Durchläufe geplant	10	
Erwartetes Ergebnis	Einwandfreies Leuchten der LEDs beim gewünschten Zustand.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gut	0
Festgestellter Fehler	Keine	

Tabelle 20: Testprotokoll T-06

ID/Bezeichnung	T-06	System überall einsetzbar
Beschreibung	Das Projekt soll so gebaut werden, dass es leicht demontiert werden kann, an einem anderen Standort wieder montiert und in Betrieb genommen werden kann.	
Testvoraussetzung	Das Gesamtsystem ist kompakt aufgebaut und leicht transportierbar.	
Durchführung	Beim Kunden montieren und demontieren und an einem anderen Standort wieder montieren.	
Durchläufe geplant	1	
Erwartetes Ergebnis	Die Montage mit dem Teleskoparm ist unproblematisch.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gut	0
Festgestellter Fehler	Keine	

9.3.3 Prüfungsprotokoll 1

Tabelle 21: Prüfungsprotokoll 1

Prüfungsprotokoll 1			
ID/Bezeichnung	Fehlerklasse	Fehlerstufe	Prüfungsdatum
T-01	Mittel	2	07.10.2020
T-02	Gut	0	07.10.2020
T-03	Gering	1	07.10.2020
T-04	Gut	0	07.10.2020
T-05	Gut	0	07.10.2020
T-06	Gut	0	07.10.2020

Die Fehler bei T-01 und T-03 müssen nun eingehend geprüft und Lösungswege erarbeitet werden. Nach der Fehlerbehebung wird eine erneute Testung vorgenommen. Dabei bekommen sie neue Bezeichnungen. T-01 wird zum T-07 und T-02 wird zum T-08.

9.3.4 Nachprüfung

Nach dem Testlauf 1 ist bei zwei *MUSS*-Zielen ein Fehler aufgetreten. Diese Fehler wurden nun behoben und mussten erneut einem Test durchlaufen.

9.3.5 Testprotokolle 2 MUSS-Ziele

Tabelle 22: Testprotokoll T-07

ID/Bezeichnung	T-07	Dauerhafte Überwachung der Welpen
Beschreibung	Das System muss die Welpen dauerhaft erkennen und überwachen.	
Testvoraussetzung	Das System ist in Betrieb und die Pixy Cam2 ist mit dem Arduino verbunden.	
Durchführung	Es wird mit dem PixyMon v2-Tool eine Verbindung mit dem Laptop hergestellt und kontrolliert, ob alle Objekte detektiert werden.	
Durchläufe geplant	5	
Erwartetes Ergebnis	Es werden keine Fehler mehr erwartet.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gut	0
Festgestellter Fehler	Keine.	

Tabelle 23: Testprotokoll T-08

ID/Bezeichnung	T-08	Fehlerfreier Betrieb über die gewünschte Überwachungszeit der Welpen
Beschreibung	Das System muss fehlerfrei funktionieren.	
Testvoraussetzung	Das Gesamtsystem ist in Betrieb und das System läuft stabil und ohne Fehler.	
Durchführung	Das System ist autonom.	
Durchläufe geplant	5	
Erwartetes Ergebnis	Da nun T-07 keine Fehler aufzeigt, ist dieser Fehler automatisch auch behoben worden.	
Testmethode	Black-Box-Test	
Fehlerklasse/ Fehlerstufe	Gut	0
Festgestellter Fehler	Keine	

9.3.6 Prüfungsprotokoll 2

Tabelle 24: Prüfungsprotokoll 2

Prüfungsprotokoll 2			
ID/Bezeichnung	Fehlerklasse	Fehlerstufe	Prüfungsdatum
T-07	Gut	0	08.10.2020
T-08	Gut	0	08.10.2020

9.3.7 Fazit Prüfung

Nach der Fehlerbehebung hat das System reibungslos funktioniert und konnte dem Kunden übergeben werden.

10 Quellenangabe und Literaturverzeichnis

10.1 Quellen aus dem Internet

Nachfolgend werden die genutzten Quellen thematisch gegliedert aufgezeigt.

Software:

<https://www.arduino.cc/en/Main/Software>

Arduino Software IDE

<https://pixycam.com/downloads-pixy2/>

PixyMon v2

<https://pixycam.com/downloads-pixy2/>

Arduino Pixy2 Bibliothek

<https://dronebotworkshop.com/pixy2-camera/>

Tutorial für Pixy Cam2

Hardware:

<https://www.bastelgarage.ch/pixy2-cmucam5-bildsensor-robot-vision?search=pixy>

Pixy Cam2

<https://www.bastelgarage.ch/arduino-uno-rev-3-smd-board-atmega328?search=arduino%20uno>

Arduino UNO

<https://www.bastelgarage.ch/4-kanal-relais-modul?search=arduino%20relais>

Relais

<https://www.bastelgarage.ch/kippschalter-on-on-mts-102?search=schalter>

Kippschalter

<https://www.digitec.ch/de/s1/product/digitus-usb-verlaengerungskabel-3m-usb-kabel-10281176>

USB Verlängerung

<https://www.digitec.ch/de/s1/product/goobay-3-v-12-v-universal-netzteil-15a-universalladegeraet-11511239>

Univeral Netzteil

Lernen / Wissen:

<https://www.arduino.cc/en/Tutorial/HomePage>

Arduino Tutorial

<https://www.youtube.com/watch?v=0wAY3DYihyg&t=3s>

Arduino Tutorial Teil 1

<https://www.youtube.com/watch?v=BKm1QFZS0m4&t=297s>

Arduino Tutorial Teil 2

<https://www.youtube.com/watch?v=eaFvQG8wrGw>

Arduino Tutorial Teil 3

<https://www.youtube.com/watch?v=t-IRfOalxhE>

Arduino Tutorial Teil 4

<https://www.youtube.com/watch?v=nf8Ugy6vLmg>

Arduino Tutorial Teil 5

<https://www.youtube.com/watch?v=QIDH2SApocM&t=129s>

Arduino Tutorial Teil 6

<https://de.wikipedia.org/wiki/Black-Box-Test>

Black-Box-Test

<https://www.youtube.com/watch?v=tnYIb5YIFMM&t=466s>

Pixy Cam Tutorial

https://www.youtube.com/watch?v=391dXDjqzXA&feature=emb_logo

Pixy Cam Tutorial

10.2 Literaturverzeichnis

Celia, I.; Manser, E. & Waldispühl, C. (2015). *Informatik für technische Kaufleute. Grundlagen mit Beispielen, Repetitionsfragen und Antworten sowie Übungen*, 5. überarbeitete Auflage. Zürich: Compendio Bildungsmedien AG

Führer, A. & Züger, R.-M. (2017). *Projektmanagement für Führungsfachleute. Management-Modul: Theorie, Methoden, Beispiele, Repetitionsfragen und Antworten*, 5. überarbeitete Auflage. Zürich: Compendio Bildungsmedien AG

Kappel, B. (2019). *Arduino. Elektronik, Programmierung, Basteln*, 2. korrigierter Nachdruck. Bonn: Rheinwerk

11 Abbildungsverzeichnis

Abbildung 1: Basis-Phasenkonzept.....	11
Abbildung 2: Projektbeteiligte.....	15
Abbildung 3: Projekttermine SOLL.....	26
Abbildung 4: Blockdiagramm Anforderung.....	29
Abbildung 5: Arduino Uno & Arduino Mega.....	33
Abbildung 6: Pixy Cam2 & M5 Camera OV 2640.....	35
Abbildung 7: ipONEgprs.....	37
Abbildung 8: Software Arduino.....	39
Abbildung 9: Software Pixy Cam2.....	39
Abbildung 10: Arduino Programmierumfeld.....	40
Abbildung 11: Pixy Cam2 Bibliothek.....	41
Abbildung 12: Code Pixy Cam2.....	41
Abbildung 13: Testprojekt Arduino & Pixy Cam2.....	42
Abbildung 14: Kontextdiagramm Vorstudie Umsetzung.....	43
Abbildung 15: Blockdiagramm Vorstudie Umsetzung.....	44
Abbildung 16: 4-Kanal Relais.....	46
Abbildung 17: ON-OFF Schalter.....	47
Abbildung 18: ON-OFF Schalter mit LEDs.....	47
Abbildung 19: SysConf ipONE Kundendaten.....	48
Abbildung 20: SysConf ipONE gprs.....	49
Abbildung 21: SysConf ipONE Ein/Ausgänge.....	50
Abbildung 22: SysConf ipONE Teilnehmer.....	51
Abbildung 23: SysConf ipONE Alarmverzögerung.....	51
Abbildung 24: SysConf ipONE Helpdesk ohne Alarm.....	52
Abbildung 25: SysConf ipONE Helpdesk mit Alarm.....	52
Abbildung 26: Kontextdiagramm Prototyp.....	54
Abbildung 27: Blockdiagramm Prototyp.....	55
Abbildung 28: Aufbau Prototyp.....	56
Abbildung 29: PixyMon Tool.....	57
Abbildung 30: Aufbau beim Kunden.....	63
Abbildung 31: Serielle Schnittstelle im IDE-Tool.....	64
Abbildung 32: Aufbau Pixy Cam2.....	65
Abbildung 33: Aufbau Hardware-Komponente.....	66
Abbildung 34: Stromlaufschema des Projekts.....	67

Abbildung 35: Fussdiagramm Software	68
Abbildung 36: Der Code Teil 1	69
Abbildung 37: Der Code Teil 2	70
Abbildung 38: Der Code Teil 3	70
Abbildung 39: Der Code Teil 3.1	71
Abbildung 40: Der Code Teil 3.2	72
Abbildung 41: Der Code Teil 3.3	73

12 Tabellenverzeichnis

Tabelle 1: Risikoeinflüsse und Beschreibung	19
Tabelle 2: Risikoanalyse	20
Tabelle 3: Massnahme.....	20
Tabelle 4: Review 1	24
Tabelle 5: Review 2	25
Tabelle 6: Projekttermine IST.....	27
Tabelle 7: Evaluation Mikrocontroller	33
Tabelle 8: Evaluation Kamera	35
Tabelle 9: Evaluation Übermittlungsgerät.....	37
Tabelle 10: Technische Daten 4-Kanal Relais.....	46
Tabelle 11: Fehlerklassifizierung.....	59
Tabelle 12: Testtabelle.....	60
Tabelle 13: Testplan	60
Tabelle 14: Kosten.....	61
Tabelle 15: Testprotokoll T-01	74
Tabelle 16: Testprotokoll T-02	75
Tabelle 17: Testprotokoll T-03	75
Tabelle 18: Testprotokoll T-04	76
Tabelle 19: Testprotokoll T-05	77
Tabelle 20: Testprotokoll T-06	77
Tabelle 21: Prüfungsprotokoll 1	78
Tabelle 22: Testprotokoll T-07	79
Tabelle 23: Testprotokoll T-08	80
Tabelle 24: Prüfungsprotokoll 2	80

13 Reflexion

In diesem Kapitel werden ausgewählte Aspekte der Abschlussarbeit reflektiert. Das Kapitel gliedert sich in Reflexionen zur Zielerreichung des Projekts, dem methodischen Vorgehen sowie zu den Lessons Learnd.

13.1 Zielerreichung des Projekts

Dieses Unterkapitel beschäftigt sich mit der allgemeinen Zielerreichung des Projekts. Dabei werden die mit dem Kunden besprochenen *MUSS*- und *KANN*-Ziele sowie Grenzen und Chancen vom Gesamtsystem erläutert (siehe Kapitel 4.6) und reflektiert. Ausserdem wird die Zielerreichung der persönlichen Ziele im Rahmen dieses Projekts überprüft.

13.1.1 Zielsetzung des Projekts

Das Projekt beschäftigt sich mit einem System zur Überwachung von Welpen. Ich habe von meinem Auftraggeber den Auftrag erhalten, ein Überwachungssystem zu realisieren und umzusetzen. Das System soll den Kunden über den gewünschten Überwachungszeitraum entlasten. Dafür wurden gemeinsam *MUSS*-Ziele definiert. Das System muss dauerhaft, über die gewünschte Zeit, einsatzbereit sein. Bei einem Ausbruch eines oder mehrerer Welpen muss der Kunde mittels einer SMS benachrichtigt werden. Dafür ist ein fehlerfreier Betrieb notwendig. Das System muss ausserdem durch einen Schalter blockierbar sein, da die Welpen nicht die ganze Zeit in der Wurfbox sind. Es wurden auch *KANN*-Ziele definiert. Jeder Welpen wird mit einem LED angezeigt, des Weiteren wird der Blockier-Modus des Schalters mit LEDs optisch angezeigt. Zusätzlich kann das System mit einem erweiterbaren Signalgeber ausgerüstet werden, wie zum Beispiel einer Blitzleuchte oder Sirene. Das System kann zudem Temperaturschwankungen erkennen und sendet dem Kunden eine SMS. Das Gesamtsystem soll so gebaut werden, dass es überall einsetzbar ist. Letzteres, die Anzeige der Welpen mittels LED sowie des Gesamtsystemzustand wurden im vorliegenden Projekt umgesetzt.

Durch das Projekt verspreche ich mir persönlich ein vertieftes Wissen im Umgang mit der ausgewählten Hard- wie auch Software. Als besondere Herausforderung erachte ich das Erlernen der Programmiersprache C sowie die Lösungsfindung im Projekt.

13.1.2 Evaluation des Projekts

Im Fach *Embedded Systeme* fand eine intensive Auseinandersetzung mit Arduino statt. Ich konnte meine im Unterricht gesammelten Erfahrungen und das erarbeitete Wissen gut im Projekt einfließen lassen. Der Aufbau der Hardware ist über die ganze Diplomzeit reibungslos verlaufen. Die Programmierung mit der Programmiersprache C stellte sich als Herausforderung heraus. Dieses Wissen musste ich mir selber aneignen und somit war der Zeitaufwand etwas grösser als geplant. Hinsichtlich meiner persönlichen Entwicklung und dem damit verbundenen Lernzuwachs würde ich in diesem Bereich den Schwerpunkt setzen. Nachträglich würde sich eine intensivere Vorbereitung oder der Besuch eines Kurses zur Programmiersprache anbieten.

13.1.3 Grenzen und Chancen des Projekts

Im Rahmen der vorliegenden Diplomarbeit konnte ein effizientes Gesamtsystem geplant und umgesetzt werden, welches auf unterschiedliche Kontexte anwendbar ist. Entstanden ist eine Grundlagedokumentation, die flexibel und erweiterbar ist. So ist das System durch weitere Signalgeber erweiterbar. Mit dem Übermittlungsgerät könnten ausserdem weitere Kriterien übermittelt werden, wie zum Beispiel Temperaturschwankungen oder eine zeitliche Begrenzung des Blockier-Modus, sollte vergessen gegangen sein, den Schalter wieder einzuschalten. Sollten mehr PINs benötigt werden, lässt sich das Arduino Uno je nach Bedarf durch ein Arduino Mega ersetzen. Die gesamte Übermittlung könnte mit einem Netzwerk realisiert werden. Es gibt eine grosse Auswahl an Arduino Shields, so sind dem System keine Grenzen gesetzt. Die verwendete Pixy Cam2 übersteigt die Anforderungen des Systems für das vorliegende Projekt. Sie besticht durch viel Potenzial und ist trotzdem kostengünstig, dieses Produkt und seine Einsatzmöglichkeiten werde ich sicherlich noch weiterverfolgen.

13.1.4 Fazit zum Projekt

In dieser Diplomarbeit ist ein System entstanden, das transparent, flexibel und erweiterbar ist. Somit sind im Rahmen dieser Arbeit die Voraussetzungen für die Weiterentwicklung kommender Projekte geschaffen worden. Die vorliegende Dokumentation dient als Grundlage und kann für weitere Projekte adaptiert werden. Insofern ist die Zielsetzung des Projekts erfüllt. Auch meine persönliche Ziele konnte ich erfüllen. Die Vertiefung mit der Hard- und Software hat dazu geführt, dass ich nun über fundiertes Wissen verfüge, welches in die praktische Arbeit einfließen konnte.

Auch der Kunde war restlos vom System überzeugt und empfand das Gesamtsystem als sinnvolle Entlastung bei der Welpenzucht.

13.2 Methodisches Vorgehen

In diesem Kapitel wird kurz erläutert, wie die Projektidee entstanden ist. Ausserdem werden die Projektphasen sowie das Zeitmanagement reflektiert.

13.2.1 Initialisierung des Projekts

Während meiner Kindheit hatte meine Familie stets Hunde. Diese waren einerseits geschätzte Mitglieder der Familie, wurden von meinem Vater aber auch für die Jagd ausgebildet und eingesetzt. Mit der Pensionierung kam bei meinem Vater die Idee auf, eine Hundezucht für eine exklusive Jagdhunderasse aufzubauen, welche in der Schweiz noch wenig etabliert ist. Die aktuelle Hündin ist eine Brandlbracke, welche im Herbst 2020 ihren zweiten Wurf hatte. Bereits beim ersten Wurf merkte die gesamte Familie, dass das Betreuen der Welpen viel Zeit beansprucht und äusserst arbeitsintensiv ist. Da meine Mutter noch berufstätig ist, kann sie meinen Vater nur bedingt unterstützen, auch wir Kinder sind bereits seit Langem ausgezogen und können nur punktuell unterstützen. Als ich beim ersten Wurf einen Nachmittag lang die Welpen betreut habe, musste ich feststellen, dass diese die volle Aufmerksamkeit auf sich ziehen. Die verantwortungsvolle Aufgabe liess keinerlei Nebenbeschäftigung zu, so wurde nichts aus der eigentlich geplanten Vorbereitung auf die Mathematiklausur. Diese Erfahrung war Ausgangspunkt für die Diplomarbeit, die Welpenüberwachung sollte die Aufzucht des zweiten Wurfs erleichtern.

13.2.2 Rückblick auf die Projektphasen

Jede Projektphase musste gründlich erarbeitet, analysiert und dokumentiert werden. Rückblickend kann ich feststellen, dass sich die einzelnen Projektphasen durch gute Planung und Reflexion reibungslos umgesetzt werden konnten. Einzig in der Grobkonzept-Phase zeigten sich hinsichtlich der Programmiersprache C grössere Schwierigkeiten als erwartet.

13.2.3 Zeitmanagement

Durch das im Vorfeld definierte Zeitmanagement und die Meilensteine konnte ich immer sehen, wo ich im Projektverlauf stehe. Die Strukturierung durch das Zeitmanagement zeigte sich bei der Umsetzung des Projekts als äusserst hilfreich und war wichtiger Orientierungspunkt, auch hinsichtlich dem Status des Gesamtprojekts. Verzögerungen und Zeitüberschreitungen konnten mittels effizienter Umsetzung anderer Aufgaben kompensiert werden. Wie bereits erwähnt, war das Erlernen der Programmiersprache C während der Grobkonzept-Phase aufwändiger als eingeplant. Ausserdem führte die Nachprüfung des Gesamtsystems dazu, dass ich das System erst nach der Fehlerbehebung am Folgetag noch einmal Prüfen konnte.

13.2.4 Methodisches Fazit

Die Idee ein Überwachungssystem für Welpen auszuarbeiten, welches an spezifische Gelingenbedingungen und Ziele gekoppelt ist, überzeugt. Die Zielerreichung des vorliegenden Projekts wird durch die planvolle methodische Umsetzung erleichtert. Besonders das Zeitmanagement dient als zentraler Orientierungspunkt und lässt erkennen, ob das Projekt gefährdet ist. Das vorausschauende Planen mittels der Projektphasen ist wichtiges Qualitätskriterium des Projekts.

13.3 Lessons learned

Lessons learned ist eine Aufzeichnung, das systematische Sammeln, Bewerten und Verdichten von Erfahrungen, Entwicklungen, Hinweisen, Fehlern und Risiken aus dem Projekt. Meine Erfahrungen werde ich nachfolgend darlegen.

13.3.1 Persönliche Zielsetzung

Durch das Erarbeiten meiner Diplomarbeit konnte ich mich im Gebiet der Mikrocomputertechnik weiterbilden und einige interessante Aspekte vertiefen. Schon während dem Studium verspürte ich eine besondere Motivation bei Lerneinheiten, welche sich mit Mikrocomputertechnik beschäftigten.

Im Laufe meiner Arbeit habe ich Möglichkeiten und Potenziale der verwendeten Hardware-Komponenten kennengelernt. Dabei wurden Technologien kombiniert, welche durch den steten technischen Fortschritt weiterentwickelt und in der Folge auch ersetzt werden. So wird das iPhone bald an sein Lebensende kommen, weil die 2G-Technologie nicht mehr länger betrieben wird. Die Einsatzmöglichkeiten und der Nutzen der Pixy Cam2 stehen wiederum erst am Anfang und wird in Zukunft die Technikwelt verändern. Mein Projekt kombiniert damit Vergangenheit, Gegenwart und Zukunft. Diese Erkenntnis ist äußerst wichtig: Es entstehen immer neue Möglichkeiten und veraltete Technologien werden ersetzt. Insofern ist es zentral, die technischen Entwicklungen zu verfolgen, um sinnvolle und effiziente Hardware-Komponenten in den Systemen zu verbauen.

Ausserdem kam es zu einer interessanten Auseinandersetzung in Bezug auf das Zusammenspiel der Hardware-Komponenten. Bei der Problemlösung musste zusätzlich ein Relais verbaut werden, um die Brücke zwischen alter und neuer Hardware zu bilden.

Nicht zuletzt konnte ich durch die Arbeit an meiner Diplomarbeit Fortschritte bezüglich meiner Software-Kenntnisse feststellen. Entstanden ist ein funktionsfähiger Code, dessen Programmierung erst durch das selbstständige Erlernen der Programmiersprache C möglich war. Es ist davon auszugehen, dass es elegantere und effizientere Lösungen für den Code gibt, jedoch funktioniert der erarbeitete Code problemlos.

13.3.2 Ausblick

Meine Diplomarbeit besticht als nützliches Grundlagendokumentation, welche sich in verschiedenen Kontexte anwenden, adaptieren und erweitern lässt. Dadurch lassen sich vielfältige Kundenwünsche erfüllen und das Potenzial des Gesamtsystems ist noch nicht erschöpft. Auf Basis meiner Erfahrungen kann eine Realisierung basierend auf dieser Grundlagendokumentation für andere Kunden umgesetzt werden.

Die Erfahrungen mit der Pixy Cam2 waren äusserst interessant, sodass ich mich im privaten Bereich auch weiterhin mit dieser Kamera beschäftigen werde und allenfalls weitere Projekte realisiere.

13.3.3 Persönliches Fazit

Im Zuge meiner Diplomarbeit habe ich mich mit der Neukombination mir bereits bekannter Komponenten beschäftigt, was eine gewisse Komplexität mit sich brachte. Entstanden ist ein Projekt, das einen spezifischen Zweck erfüllt und individuell auf den Kunden zugeschnitten ist. Die dabei entstandene Grundlagendokumentation lässt sich im Rahmen unterschiedlicher Projekte flexibel einsetzen.

Erstmals habe ich mein Wissen aus verschiedenen Gebieten und zu verschiedenen Komponenten für ein Gesamtsystem effizient kombiniert und aktiv nach Lösungswegen gesucht, um eine Verbindung zwischen den Einzelkomponenten zu schaffen. Dies war teilweise herausfordernd und hat aufgezeigt, dass sich der technologische Fortschritt sich unaufhörlich wandelt. Insofern muss man am Ball bleiben und sich ständig mit Neuerungen und Verbesserungen auseinandersetzen.

Dabei fasziniert mich besonders die Mikrocomputertechnologie, sie entwickelt sich ständig weiter und birgt zahlreiche Möglichkeiten. Daher werde ich mich auch in Zukunft eingehend mit der Thematik beschäftigen. Es sind bereits erste Ideen für weitere Einsatzmöglichkeiten entstanden, beispielsweise ein System zur Vertreibung von Katzen auf dem eigenen Grundstück durch ausgelöste Wasserpistolen. Ich bin mit meinen persönlichen Lernfortschritten, welche ich im Zusammenhang mit der vorliegenden Diplomarbeit gemacht habe, zufrieden und bin motiviert, mich auch weiterhin mit der Thematik zu beschäftigen, sei dies im beruflichen oder auch im privaten Bereich.

13.4 Verdankung

Florian Müller

Ich möchte mich bei meinem ehemaligen Chef bedanken. Durch seine Unterstützung wurde es mir ermöglicht, diese Weiterbildung überhaupt anzugehen.

Andreas Böhlen

Andreas ist ein Disponent in meiner Firma. Ich konnte diverses Kleinmaterial und das Übermittlungsgerät bei ihm beziehen.

Angela Aegerter

Meine Schwester hat mich durch mein Studium begleitet und hat mich in dieser Zeit besonders beim Korrekturlesen meiner Arbeiten unterstützt.

Erika und Hermann Aegerter

Meine Eltern haben mich durch das Studium begleitet und ich konnte mit Hilfe von ihnen meine Diplomarbeit realisieren.

Mika Lao und Philipp Haefeli

Mika und Philipp sind ehemalige Studenten der TEKO. Sie haben mich motiviert ein Studium zu beginnen und haben mich im Fach Mathematik sehr unterstützt.

Anna Weyh

Anna hat mich bei der Finalisierung der Diplomarbeit unterstützt und die letzte Kontrolle der Arbeit umgesetzt.

13.5 Eidesstattliche Erklärung

Ich bestätige, dass ich die vorliegende Diplomarbeit selbstständig verfasst und alle benutzten Quellen gekennzeichnet habe. Diese Arbeit wurde weder in gleicher noch in ähnlicher Form bereits einer Prüfungskommission vorgelegt.

Boll, 01.11.2020

Ort, Datum

J. Müller

Unterschrift des Verfassers