

Dokumentation zu „LeafySan“

Ein Projekt für den bundesweiten Wettbewerb „Invent A Chip“ 2017 von Florian Winkler.

1 LeafySan.....	2
1.1 Modell und Projektaufbau.....	2
1.2 Frontend.....	3
1.3 Stromversorgung.....	4
1.4 Unterstützer.....	4
2 Entwicklungsprozess.....	5
3 Kosten.....	8
4 Architektur des FPGA-Chips.....	9
4.1 Auslesen der Sensoren.....	10
4.1.1 Temperatursensor und CO2-Sensor.....	10
4.1.2 Helligkeitssensor und Feuchte-Sensor.....	11
4.2 Steuerung der Aktoren.....	14
4.2.1 Beleuchtung.....	14
4.2.2 Bewässerung.....	15
4.2.3 Belüftung.....	16
4.2.4 Heizung.....	17
4.3 Ressourcennutzung.....	17
4.4 Stromverbrauch.....	18
4.5 Testbenches.....	18
5 Schlusswort.....	19

1 LeafySan

LeafySan - ein vollautomatisiertes Gewächshaus gesteuert und überwacht durch einen FPGA-Chip. Dieses Projekt wurde im Zuge des bundesweiten Nachwuchswettbewerbes "Invent a Chip" im Jahr 2017 erstellt. Mithilfe der Hardwarebeschreibungssprache VHDL und einem DE2-Board von Altera wurde das Auslesen der Sensoren und Steuern der Aktoren umgesetzt. Des weiteren werden Sensordaten an einen Raspberry Pi gesendet, welcher diese Daten speichert. Die Daten können auf mobilen Geräten wie Tablets oder Smartphones sowie Laptops im Browser abgerufen werden.

FPGA Entwicklungsboard:

- Altera DE2-115 Board

verwendete Sensoren:

- digitaler Helligkeitssensor (I²C) (http://wiki.seeed.cc/Grove-Digital_Light_Sensor/)
- digitaler Feuchte-Sensor (I²C) (<https://www.tindie.com/products/miceuz/i2c-soil-moisture-sensor/>)
- analoger Temperatursensor (PT1000 Thermistor) (<https://www.reichelt.de/Sensorik-SMD-bedrahtet-/PCA-1-1505-10/3/index.html?ACTION=3&LA=446&ARTICLE=85054&GROUPID=8097&artnr=PCA+1.1505+10&SEARCH=PCA%2B1.1505%2B10>)
- CO2 Infrarot Sensor (0~5000 ppm) (http://www.komputer.de/zen/index.php?main_page=product_info&cPath=24&products_id=424)

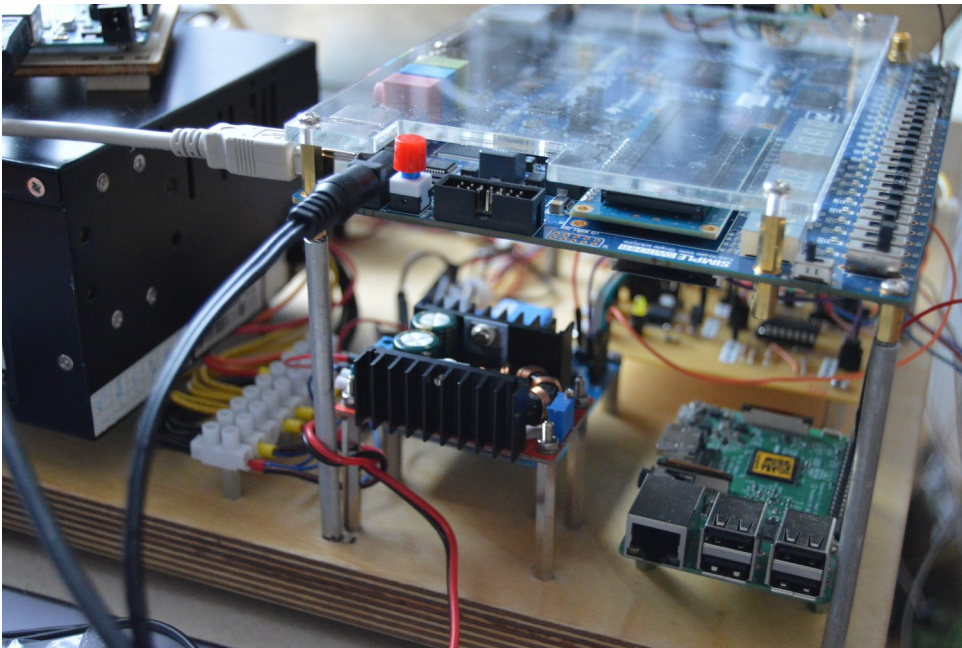
verwendete Aktoren:

- Lüfter (12V, 40x40mm) (<https://www.reichelt.de/Luefter/FAN-ML-4010-12-S/3/index.html?ACTION=3&LA=446&ARTICLE=110411&GROUPID=7775&artnr=FAN-ML+4010-12+S&SEARCH=FAN-ML%2B4010-12%2BS>)
- Wasserpumpe
- 2x Heizung (12V, 12W, 77x110mm) (<https://www.conrad.de/de/polyester-heizfolie-selbstklebend-12-vdc-12-vac-12-w-schutzart-ipx4-l-x-b-110-mm-x-77-mm-thermo-532878.html>)
- 2x LED-Streifen (jeweils 25cm Länge) (<http://www.ledlager.de/led-streifen>)

1.1 Modell und Projektaufbau

Die Elektronik wurde auf einer Holzplatte mit Schrauben und Stelzen befestigt, damit bei einem Wasserleck nicht die sich auf dem Boden befindliche Elektronik beschädigt wird. Um die Funktionsweise der einzelnen Sensoren und Aktoren zu demonstrieren, wurde ein "Gewächshaus" angefertigt, das aus einem Aluminiumboden und einer Acrylglas-Haube besteht. Darin befindet sich Blumenerde, die von einem kleinen Schlauchsystem bewässert wird, sowie

der Bodenfeuchte-Sensor, der die Feuchtigkeit des Boden nicht-kapazitiv misst. Außerdem wurde die Beleuchtung und Belüftungsmechanik (ein Elektromagnet öffnet/schließt eine Klappe, um den Gasaustausch mit der Außenluft zu ermöglichen) im Gewächshaus verbaut. Damit das Gewächshaus erwärmt werden kann, steht es auf einer Aluminiumplatte, welche durch zwei kleine 12V Heizplatten erwärmt werden kann.



Die noch notwendige Mess- und Steuerelektronik befindet sich auf einer Lochrasterplatine unterhalb des DE2-Boards, wo sich auch eine Relaiskarte mit vier steuerbaren Relais sowie Raspberry Pi befinden.

1.2 Frontend

Das DE2-Board sendet über ein UART Interface die aktuellen Sensorwerte an den Raspberry Pi 3. Der Mikrocomputer speichert aller fünf Sekunden die Sensorwerte des aktuellen Tages in einer CSV-Datei. Somit dient der Raspberry Pi als "Archiv" der Standortbedingungen der Pflanze und soll dem Nutzer darüber Aufschluss geben.

An dieser Stelle großen Dank an Phil Martins Tutorial "Using your new Raspberry Pi 3 as a WiFi Access Point" (<https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/>)

Nachdem der Pi als Access Point eingerichtet wurde, muss der Javascript Code des Servers mithilfe von NodeJS ausgeführt werden, damit das Frontend über den Pi erreichbar ist. Wenn alles richtig eingestellt und ausgeführt wurde, ist die Seite nach Verbinden mit dem Access Point des Pi's unter `http://192.168.2.1:8000` aufrufbar.

Das Design der Seite wurde auch für mobile Geräte, wie z. B. Tablets oder Smartphone jeglicher Bildschirmgrößen, optimiert.



1.3 Stromversorgung

Damit das DE2-Entwicklerboard von Altera, sowie der Raspberry Pi 3 und die Aktoren mit ausreichend Strom versorgt werden, wurde ein altes ATX Computer-Netzteil modifiziert, um die Stromversorgung zu garantieren. Damit man die Versorgungsspannungen 3,3 V, 5 V und 12 V nutzen kann, musste jedoch die "Power Supply On" (Violett) Leitung mit Ground (Schwarz) verbunden werden, damit die 3,3 V (Orange), 5 V (Rot) und 12 V (Gelb) Schienen aktiv sind. Außerdem wurde ein 11W Leistungswiderstand mit 4,7 Ohm direkt innerhalb des Netzteilgehäuses fixiert und mit Ground und 5 V verbunden, damit eine Grundbelastung von ca. 1 A vorherrscht. Dies wird empfohlen, um das Netzteil richtig zu nutzen.

Für folgende Komponenten wurden die benannten Versorgungsspannungen benötigt:

3,3 V	5 V	12 V
Temperatursensor	Raspberry Pi 3	LED-Beleuchtung
Feuchte-Sensor	Relais	Belüftung (Lüfter + Elektromagnet)
Helligkeitssensor	CO ₂ -Sensor	Bewässerungspumpe
Relaisansteuerung		Heizung

Teilweise wurden die 3,3 V auch vom DE2-Board abgegriffen. Dieses wird wiederum vom Netzteil mit 12V betrieben.

1.4 Unterstützer

- Creative Factory GmbH Großenhain (<http://creative-factory-gmbh.de/>)
- Sebastian Fischer, MdL (CDU)
- Martin Eichler

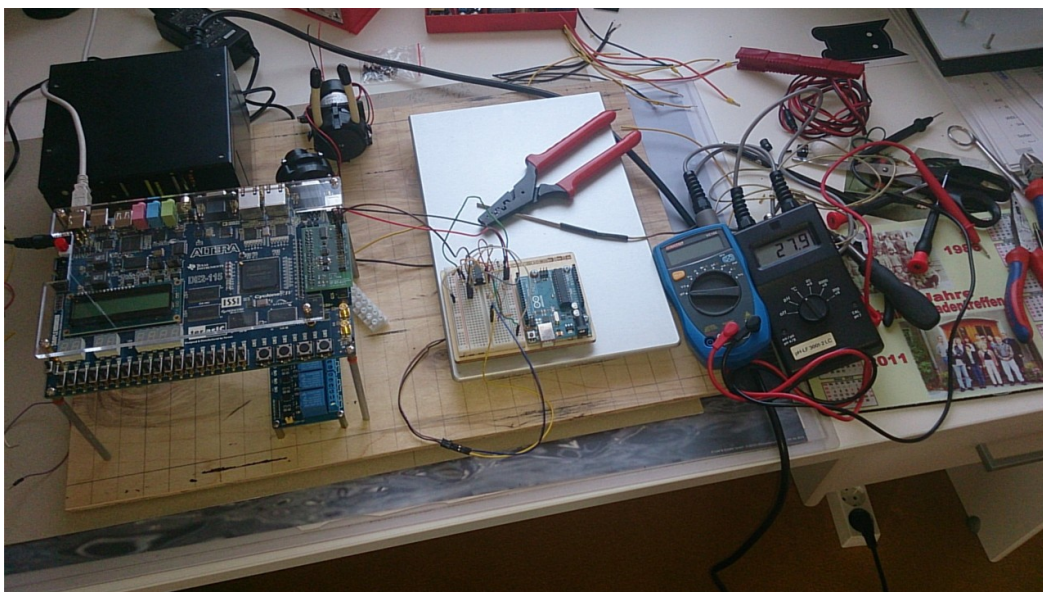
2 Entwicklungsprozess

17.05. Meilenstein 1: Die erste Aufgabe, die im Zuge des ersten Meilensteins umgesetzt wurde, war das Erstellen der Projektplanung und einer Liste mit den benötigten Sensoren und Aktoren.

27.05. Bauteilauswahl: Danach wurde die Tauglichkeit der Komponenten mit dem Betreuer besprochen. Einige dieser Komponenten wurden dann online bestellt.

03.06. Lieferung der Bauteile: Zu diesem Zeitpunkt sind bereits fast alle bestellten Bauteile angekommen und es wurde mit der Verschaltung begonnen.

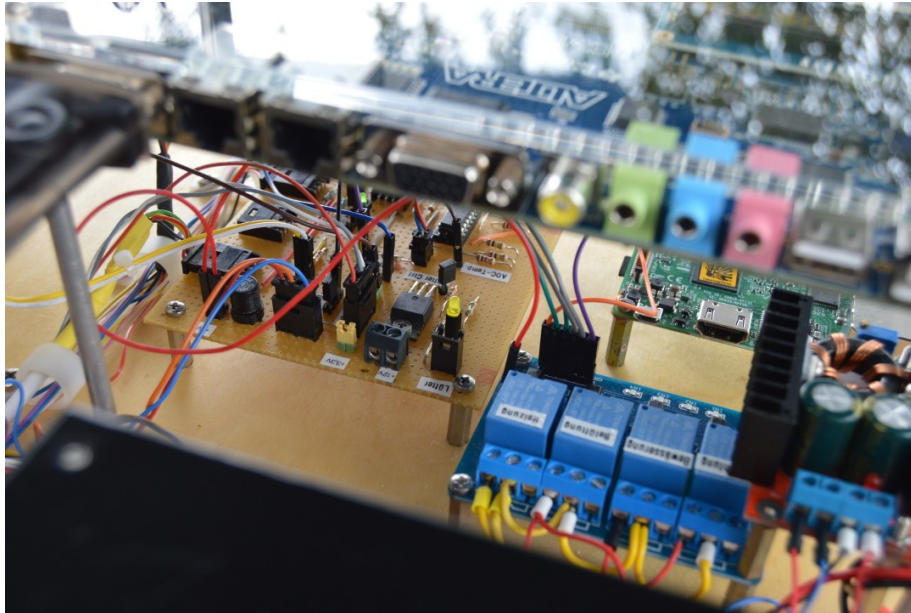
04.06. Verschaltung: Damit der Platin-Messwiderstand korrekt verstärkt wird, wurde die Instrumentationsverstärker-Schaltung mit dem Mentor besprochen und auf einem Breadboard realisiert, um den Thermistor zu kalibrieren bzw. die Umrechnung des eingelesenen Wertes zu ermitteln. An dieser Stelle entstand das erste Problem, da die Relais auf dem Relaismodul eine Spannung von 5 V benötigten, um zu schalten.



15.06. Computernetzteil: Die Lösung des oben angesprochenen Problems war die "Zweckentfremdung" eines alten Computernetzteils. Daraus kann man sowohl 5 V als auch 3,3 V und 12 V abgreifen. Die Versorgung der Relais sowie der zukünftigen Peripherie war ab diesem Punkt kein Problem mehr.

18.06. Grundplatte: Als Fundament für Heizplatte, Gewächshaus, Netzteil und Elektronik dient eine mehrschichtige Sperrholzplatte. Auf dieser wurde angefangen, die einzelnen elektronischen Komponenten, wie zum Beispiel das DE2-Board, der Raspberry Pi oder das Relaismodul, anzuordnen und auf Stelzen zu verschrauben. Außerdem wurde die Heizplatte installiert und in Betrieb genommen.

20.06. Verbindungsplatine: Da die Messelektronik bisher auf einem Breadboard zusammengesteckt wurde, erfolgte nun die Übertragung der Instrumentationsverstärker-Schaltung auf die Lochrasterplatine und wurde verlötet. Später wurden auch die Steckplätze für den I²C Bus darauf verbunden und mit Pull-Up Widerständen bestückt.



22.06. Probleme mit I²C: Das Auslesen der digitalen Sensoren, die über I²C mit dem DE2-Board kommunizieren sollen, entpuppte sich als schwierige Angelegenheit. Deshalb wurde die Hilfe bezüglich des `i2c_master` Interfaces in Anspruch genommen.

03.07. Konstruktion: Das Gewächshausdach wird aus Polystyrolplatten und Metallschienen eigenhändig zusammengebaut. Mithilfe eines Holzgerüsts, welches von einem ortsansässigen Tischlers angefertigt wurde, konnte die Polystyrolplatte in Form gebracht werden. Leider erwies sich das Material als wenig geeignet, da es bei Belastung zur Rissbildung neigte.

15.07. Neukonstruktion: Nach Bestellung einer robusteren 2 mm Plexiglasplatte, wurde das "Glasdach" erneut konstruiert.

18.07. Probleme mit den Sensoren: Nachdem das DE2-Board während einer aktiven I²C Kommunikation ausgeschalten wurde, neigte der Feuchte-Sensor dazu, seine Adresse zu ändern. Dies erschwerte die Arbeit mit dem Sensor, da die Adresse mit einem Arduino Uno auf den Standardwert zurückgesetzt werden musste. Außerdem schwankten die eingelesenen Kanalwerte des Helligkeitssensors und die Kalkulation des Lux-Wertes gestaltete sich schwierig.

25.07. Stabile Werte: Die beiden digitalen Sensoren wurden voneinander getrennt, sodass der Helligkeitssensor auf einem anderen Bus als der Feuchte-Sensor kommunizierten. Des weiteren wurden die Widerstandsgrößen von 10 kOhm auf 4,7 kOhm verkleinert. Diese Maßnahmen führten dazu, dass beide Sensoren endlich stabile und realitätsnahe Werte lieferten. Später stellte sich heraus, dass die Werte des Feuchte-Sensor noch stabiler wurden, wenn man anstelle von 4,7 kOhm nur 1 kOhm Pull-Up Widerstände verwendet.

02.08. Web Interface: Aufgrund dessen, dass das Lesen der Sensorwerte endlich funktionierte, widmete man sich dem Erstellen des Web Interfaces und der Darstellung der Werte in einem Diagramm. Außerdem wurde der Raspberry Pi zu einem Access Point konfiguriert, sodass der HTTP-Server über WLAN erreicht werden konnte.

25.08. Inbetriebnahme: Freundlicherweise unterstützte mich die Creative Factory GmbH aus Großenhain bei der Konstruktion einer Aluwanne. Als diese fertig war, wurden alle Komponenten miteinander verkabelt. Damit konnte das gesamte System in Betrieb genommen werden.

27.08. Temperaturmessung: Da die Werte des Temperatursensors an der Heizung stark schwankten (aufgrund der guten Wärmeleitfähigkeit der Heizplatte), wurde die Temperaturmessung auf den internen Feuchte-Sensor übertragen, der die vorherrschende Temperatur innerhalb des Gewächshauses misst und diesen über I²C an das DE2-Board übertragen kann. Da der Platz auf dem LC-Display nicht mehr für das Anzeigen aller Sensorwerte ausreichte, wurde die Temperatur an der Heizplatte auf die 7-Segment Anzeigen ausgelagert.

28.08. Dokumentation: Es wurde begonnen die Dokumentation über die Funktionsweise, den Aufbau und die Umsetzung des Projektes zu schreiben. Außerdem wurden auch Tests in der Natur durchgeführt, ein Video zur Demonstration der Funktionstüchtigkeit gedreht und zahlreiche Bilder erstellt.

03.09. Abgabe



3 Kosten

Nach einem Zeitungsartikel, bot Herr Sebastian Fischer (Mitglied des sächsischen Landtages) seine Unterstützung an und spendete dem Projekt einen Geldbetrag von 150 €. Außerdem ermöglichte mir die Creative Factory GmbH Großenhain die Konstruktion des Aluminiumbodens.

Die Kosten, die durch das Kaufen von Sensoren, Aktoren, Kabeln und Konstruktionsmaterialien verursacht wurden, sind hier einmal zusammengefasst dargestellt:

Modellkonstruktion	Anzahl	Preis pro Stück	Gesamtpreis	geliefert von / gekauft bei
Kleine Polystyrol-Platte 2mm	2	5,99 €	11,98 €	toom Baumarkt
Große Polystyrol-Platte 2mm	1	-	13,99 €	toom Baumarkt
Edelstahlwinkel	1	4,99 €	4,99 €	toom Baumarkt
Nieten	250	-	3,99 €	toom Baumarkt
Acrylglas GS 3mm 30x90cm	1	19,19 €	19,19 €	plattenzuschnitt24
Elektronische Komponenten	Anzahl	Preis pro Stück	Gesamtpreis	geliefert von / gekauft bei
LED-Streifen rot 25cm	1	2,25 €	2,25 €	LED-Lager
LED-Streifen blau 25cm	1	2,75 €	2,75 €	LED-Lager
Polyester-Heizfolie 12W	2	7,50 €	15,00 €	voelkner
Kabelset für Raspberry, 20 Stück	1	6,95 €	6,95 €	voelkner
Raspberry Pi 3 Model B	1	41,34 €	41,34 €	voelkner
Lochrasterplatine 150x100mm	1	1,67 €	1,67 €	reichelt
Lüfter 40x40mm 12V	1	2,31 €	2,31 €	reichelt
IC-Sockel 16-polig	2	0,21 €	0,42 €	reichelt
Platin-Chip-Temperatursensor, 1 kOhm	2	1,51 €	3,02 €	reichelt
Keramikkondensator 100nF	2	0,04 €	0,08 €	reichelt
Drahtwiderstand 11W 4,7 Ohm	2	0,52 €	1,04 €	reichelt
Quad Operationsverstärker	2	1,01 €	2,02 €	reichelt
Grove digitaler Lichtsensor	1	7,14 €	7,14 €	reichelt
Verschiedene Kohleschichtwiderstände	40	0,03 €	1,12 €	reichelt
analoger CO2 Sensor	1	39,80 €	39,80 €	Komputer.de
Bodenfeuchte-Sensor	1	18,88 €	18,88 €	tindie

ZWISCHENSUMME = 181,05 €

Zulieferer	Versandkosten
toom Baumarkt	0,00 €
plattenzuschnitt24	6,95 €
LED-Lager	3,99 €
voelkner	0,00 €
reichelt	4,71 €
Komputer.de	3,50 €
tindie	4,29 €

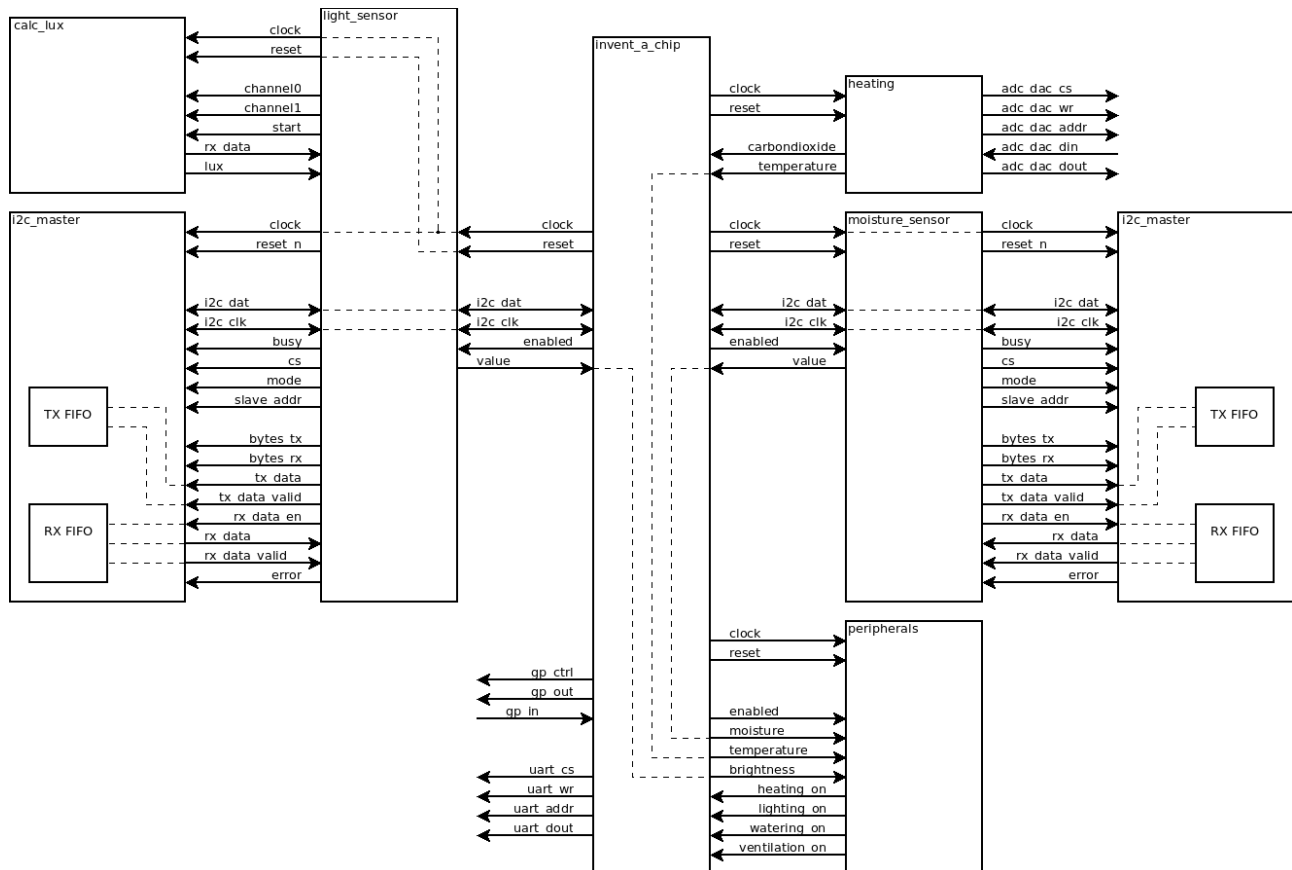
SUMME = 23,44 €

GESAMTKOSTEN = 204,49 €

4 Architektur des FPGA-Chips

Mithilfe der Hardwarebeschreibungssprache VHDL wurde die Architektur des Chips auf dem DE2-Entwicklungsboards beschrieben. Als IDE wurde Altera's Quartus verwendet. Simulationen wurden mit dem Programm ModelSim durchgeführt. Im Ordner `src` findet man die entsprechenden Projektdateien sowie den Quellcode.

Folgendes Blockschaltbild verdeutlicht leicht vereinfacht den Datenaustausch der einzelnen Entities:



Die zentrale Entity `invent_a_chip` steuert mithilfe der `gp_*` Signale die Peripherie über ein Relaismodul. Die Kommunikation über den Datenbus I2C findet auch über GPIO-Pins mithilfe der `gp_*` Signale statt. Dabei ist zu beachten, dass diese sowohl zum Schreiben als auch zum Lesen verwendet werden müssen, um die Sensoren richtig anzusteuern und auszulesen.

4.1 Auslesen der Sensoren

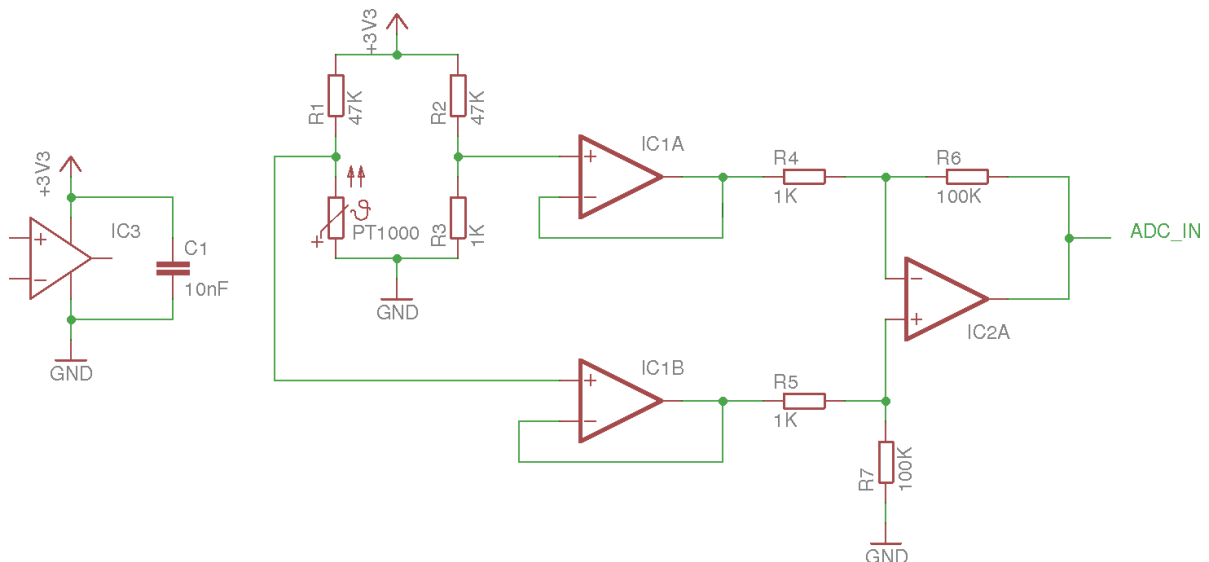
4.1.1 Temperatursensor und CO2-Sensor

VHDL-Implementierung: `src/vhdl/modules/adc_sensors.vhdl`

Sowohl Temperatur- als auch CO2-Sensor liefern dem DE2-Board einen analogen Wert, das heißt eine Spannung, die von dem Extension Board in einen digitalen Wert umgewandelt werden kann. Außerdem werden alle analog eingelesenen Werte über einen Zeitraum von fünf Sekunden addiert und daraus wird ein Mittelwert gebildet. Dadurch treten keine großen, ungewollten Schwankungen auf.

Temperatursensor

Bei dem Temperatursensor handelt es sich um einen PT1000 Thermistor, welcher bei 0 °C einen Widerstand von 1 kOhm hat. Dieser Thermistor hat einen positiven Temperaturkoeffizienten, was bedeutet, dass bei einer Temperaturerhöhung sich der Widerstand des Messwiderstandes vergrößert. Diese Änderung des Widerstandes ist annähernd linear und kann mithilfe der folgenden Verschaltung ausgelesen werden.



Der Platin-Messwiderstand (PT1000) befindet sich in einer Wheatstone'schen Messbrücke, wobei $R1 = R2$ ist. Bei einer Temperatur von 0 °C beträgt die Spannungsdifferenz der Messbrücke 0 V, weil der Widerstand des Thermistors gleich $R3$ (1 kOhm) ist. Insofern eine höhere Temperatur erreicht wird, erhöht sich die Spannungsdifferenz, welche durch den Operationsverstärker (IC2A) aufgrund der Widerstandsanordnung ($R4$ und $R5$: 1 kOhm, $R6$ und $R7$: 100 kOhm) hundertfach verstärkt wird. Die Operationsverstärker (IC1A, IC1B) dienen zur Impedanzwandlung. Alle verwendeten Operationsverstärker werden mit einer durch einen Kondensator (C1) entkoppelten 3,3 V Spannung versorgt (IC3). Die gesamte Schaltung bezeichnet man als Instrumentationsverstärker-Schaltung.

Nach der hundertfachen Verstärkung kann die resultierende Spannung durch einen ADC (**A**nalog-**D**igital **C**onverter) in einen digitalen Wert umgewandelt werden. Die Auflösung dieses ADC sind 12-bit, das heißt bei Null-Potenzial beträgt der Wert 0, bei einer Eingangsspannung von 3,3 V ist der digitale Wert 4095. Nach einer Messung bei Zimmertemperatur wurde der lineare Zusammenhang zwischen digitalem Wert und Temperaturwert in Celsius ermittelt. Mithilfe dieses Codes wird der eingelesene digitale Wert in die Einheit Temperatur umgewandelt:

```

heating_temp_next <= resize(shift_right(unsigned(temp_value) * 74043 + 131072, 18),
heating_temp'length);
-- Beispiel: temp_value = 1200
-- 1200 * 74043 + 131072 = 88982672
-- 88982672 >> 18 = 339 (entspricht der Division mit dem Divisor 2^18 [= 262144])
-- 339 entsprechen 33,9 °C

```

CO2-Sensor

Technische Details: <http://www.futurlec.com/Datasheet/Sensor/MH-Z14.pdf>

Der CO2-Sensor hat einen Messbereich von 0-5000 ppm (**p**arts **p**er **m**illion). Dies entspricht einer Konzentration von 0 bis 0.5 Vol.-% Kohlenstoffdioxid in der Luft. In der Außenluft liegt dieser Wert bei ungefähr 350 ppm, in einem Wohnraum zwischen 1000 und 2000 ppm.

Da der CO2-Sensor eine analoge Spannung zwischen 0.4 V und 2.0 V ausgibt, wird der Wert wie folgt im FPGA umgerechnet:

```

co2_next <= resize(shift_right((unsigned(co2_value) - 496) * 82539, 15), co2'length);

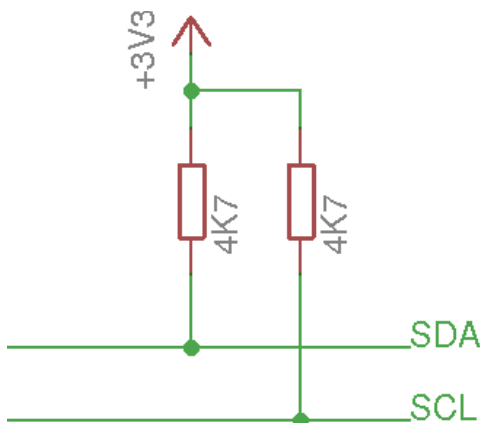
```

Die Konzentration von CO2 in der Luft des Gewächshauses soll Rückschlüsse auf die Photosyntheseleistung der Pflanze ermöglichen, da bei diesem für die Pflanze lebenswichtigen Prozess Kohlenstoffdioxid benötigt wird. Bei einer guten Photosyntheseleistung müsste der Gehalt von CO2 innerhalb des abgeschlossenen Gewächshauses sinken. Damit wieder CO2 in die Luft des Gewächshauses gelangt, wird jede viertel Stunde "belüftet", um das Stattfinden der Photosynthese zu ermöglichen.

4.1.2 Helligkeitssensor und Feuchte-Sensor

Sowohl der Helligkeitssensor als auch der Feuchte-Sensor kommunizieren über den weit verbreiteten seriellen Bus I²C. Dafür werden nur zwei Kabel (VCC und GND ausgenommen) benötigt, um die beidseitige Kommunikation vom Master (FPGA) zum Slave (Sensor) oder andersherum zu ermöglichen. Beide Sensoren kommunizieren über die Taktfrequenz 400 kHz (fast mode) und werden mit 3,3 V versorgt.

Außerdem muss der Clock- und Daten-Kanal mit einem Pull-Up Widerstand auf 3,3 V "gezogen" werden. Geeignete Widerstandsgrößen sind liegen ungefähr zwischen 2,2 kOhm und 10 kOhm. Folgender Schaltplan veranschaulicht die Verschaltung des Bus:



Helligkeitssensor

Technische Details: https://raw.githubusercontent.com/SeeedDocument/Grove-Digital_Light_Sensor/master/res/TSL2561T.pdf

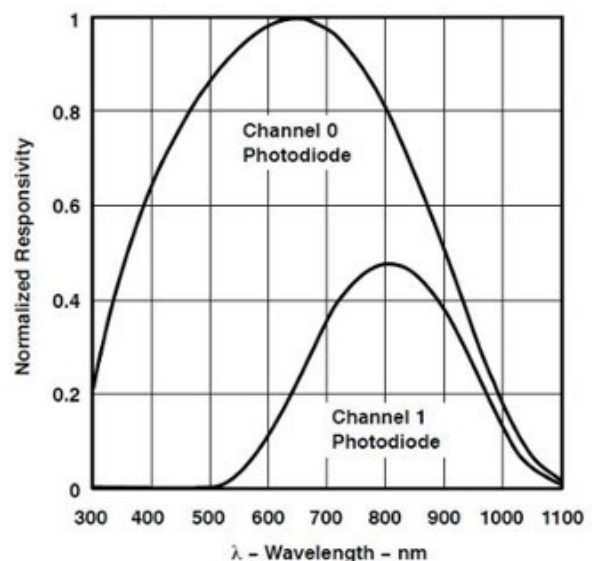
VHDL-Implementierung: `src/vhdl/modules/light_sensor.vhdl`

Der Helligkeitssensor besitzt die Slave-Adresse 0x29 und wird jede Sekunde ausgelesen. Damit der Sensor einen Wert zurückgeben kann, muss dieser zuerst "eingeschaltet" werden. Daraufhin wird eine Einstellung für das Auslesen geschickt und einen Moment gewartet, bis der Lichtsensor einen sinnvollen Wert an das DE2-Board schicken kann. Insofern alle Werte ausgelesen wurden, wird der Sensor wieder ausgeschaltet, um Strom zu sparen.

Die Tabelle soll den Kommunikationsablauf näher verdeutlichen:

Register (<i>Address-Byte</i>)	Kommando (<i>Daten-Byte</i>)	Beschreibung
REG_CONTROL (0x80)	POWER_ON (0x03)	aktiviert den Sensor
-	-	Warte 400 ms, damit der Sensor booten kann
REG_TIMING (0x81)	HIGH_GAIN, INT_101 (0x11)	Setzt Verstärkung auf 16x und Integrationszeit auf 101 ms
-	-	Warte 800 ms, damit der Sensor integrale Werte liefern kann
REG_CHANNEL0L (0x8C)	<i>Kein Kommando</i>	Lesen des tiefen Bytes des ersten Kanals
REG_CHANNEL0H (0x8D)	<i>Kein Kommando</i>	Lesen des hohen Bytes des ersten Kanals
REG_CHANNEL1L (0x8E)	<i>Kein Kommando</i>	Lesen des tiefen Bytes des zweiten Kanals
REG_CHANNEL1H (0x8F)	<i>Kein Kommando</i>	Lesen des hohen Bytes des zweiten Kanals
-	-	Lux-Wert aus ausgelesenen Kanalwerten kalkulieren
REG_CONTROL (0x80)	POWER_OFF (0x00)	Deaktiviert den Sensor

Die Integrationszeit wird benötigt, damit der Helligkeitssensor aus den analogen Werten der Photowiderstände digitale Werte an das interne ADC-Register sendet, die integer sind. Außerdem müssen die Kanalwerte der unterschiedlich empfindlichen Photowiderstände in einen Lux-Wert umgewandelt werden, der Auskunft über die Helligkeit gibt - unabhängig von der Wellenlänge des Lichts.



Je nach Verhältnis der beiden Werte werden die einzelnen Kanalwerte, die zuvor je nach Integrationszeit mit einem Faktor (bei 101 ms Integrationszeit circa `3,975`) skaliert werden, mit unterschiedlichen Koeffizienten multipliziert.

Um einen kritischen Pfad auszuschließen, wird der Lux-Wert in einer eigenen Entity über fünf Taktzyklen berechnet.

Feuchte-Sensor

technische Details: https://raw.githubusercontent.com/SeeedDocument/Grove-Digital_Light_Sensor/master/res/TSL2561T.pdf

VHDL-Implementierung: *src/vhdl/modules/moisture_sensor.vhdl*

Der Feuchte-Sensor wird jede Sekunde von dem DE2-Board ausgelesen und gibt einen 16-bit Wert an den Master zurück, der für gewöhnlich zwischen 300 und 700 liegt. Leider gibt es keine vergleichbare Einheit, die eine Aussage über die Bodenfeuchte geben kann, weshalb der Nullpunkt bei dem Wert 370 angenommen wird und der höchste Wert bei 600 liegen soll, was 100 % entsprechen soll. Folgende Wertetabelle zeigt gemessene Referenzwerte:

Zustand	Wert (min.)	Wert (max.)
In Luft	300	320
In trockener Erde	365	375
In normaler Erde	400	440
In feuchter Erde	460	480
In Trinkwasser	590	600
In gesättigter Kochsalz-Lösung	670	680

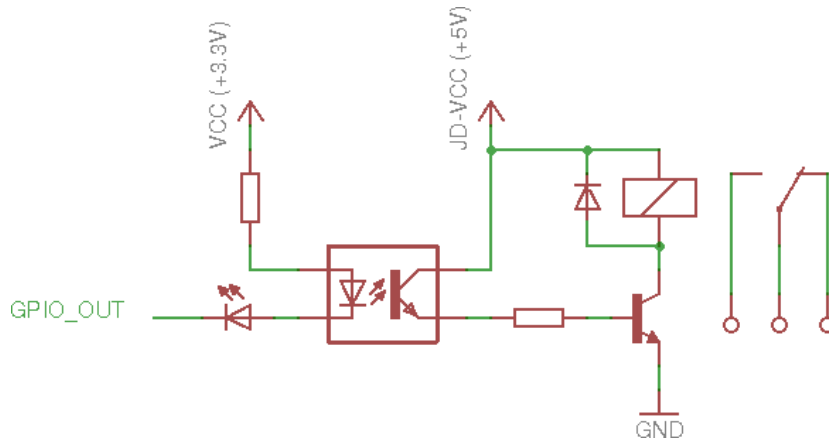
Der Ablauf der Kommunikation zwischen Sensor und DE2-Board beginnt mit dem Senden eines Reset-Signales zur Initialisierung. Dieses Signal wird nur beim "ersten Start" oder nach Betätigen des Reset-Knopfes gesendet. Jede folgende Kommunikation besteht nur aus dem Senden des zu lesenden Registers (1 Byte) und das Empfangen des Registerwertes (2 Byte).

Da die Implementierung in der Hardwarebeschreibungssprache VHDL sich an den vom Hersteller vorgeschlagenem Beispielcode (<https://github.com/Miceuz/i2c-moisture-sensor/blob/master/README.md#arduino-example>) für einen Arduino orientiert, wird zwischen dem Senden des Registerbytes und dem Empfangen des Wertes 20 ms gewartet siehe:

```
unsigned int readI2CRegister16bit(int addr, int reg) {
    Wire.beginTransaction(addr);
    Wire.write(reg);
    Wire.endTransmission();
    delay(20);
    Wire.requestFrom(addr, 2);
    unsigned int t = Wire.read() << 8;
    t = t | Wire.read();
    return t;
}
```

4.2 Steuerung der Aktoren

Zum Ansteuern der Aktorik, die eine Spannung von 12 V benötigen, wurde eine Relaismodul mit vier steuerbaren Relais verwendet. Die Relais an sich benötigen eine Spannung von 5 V für den eingebauten Elektromagneten. Damit dies sich dennoch mit der Betriebsspannung des DE2-Boards 3,3 V verträgt, ist auf dem Modul der Relaisschaltkreis mit einem Optokoppler vom Steuerschaltkreis getrennt. Hier der Schaltplan:



Der Schaltplan wurde aus den Angaben des Herstellers erstellt und enthält deshalb keine Widerstandsgrößen, weil keine Angaben in den dazugehörigen Datenblättern gefunden werden konnten. Damit der Elektromagnet im Relais anzieht, muss logisch '0' ausgegeben werden, bei logisch '1' öffnet das Relais. Die Steuerung ist somit invertiert.

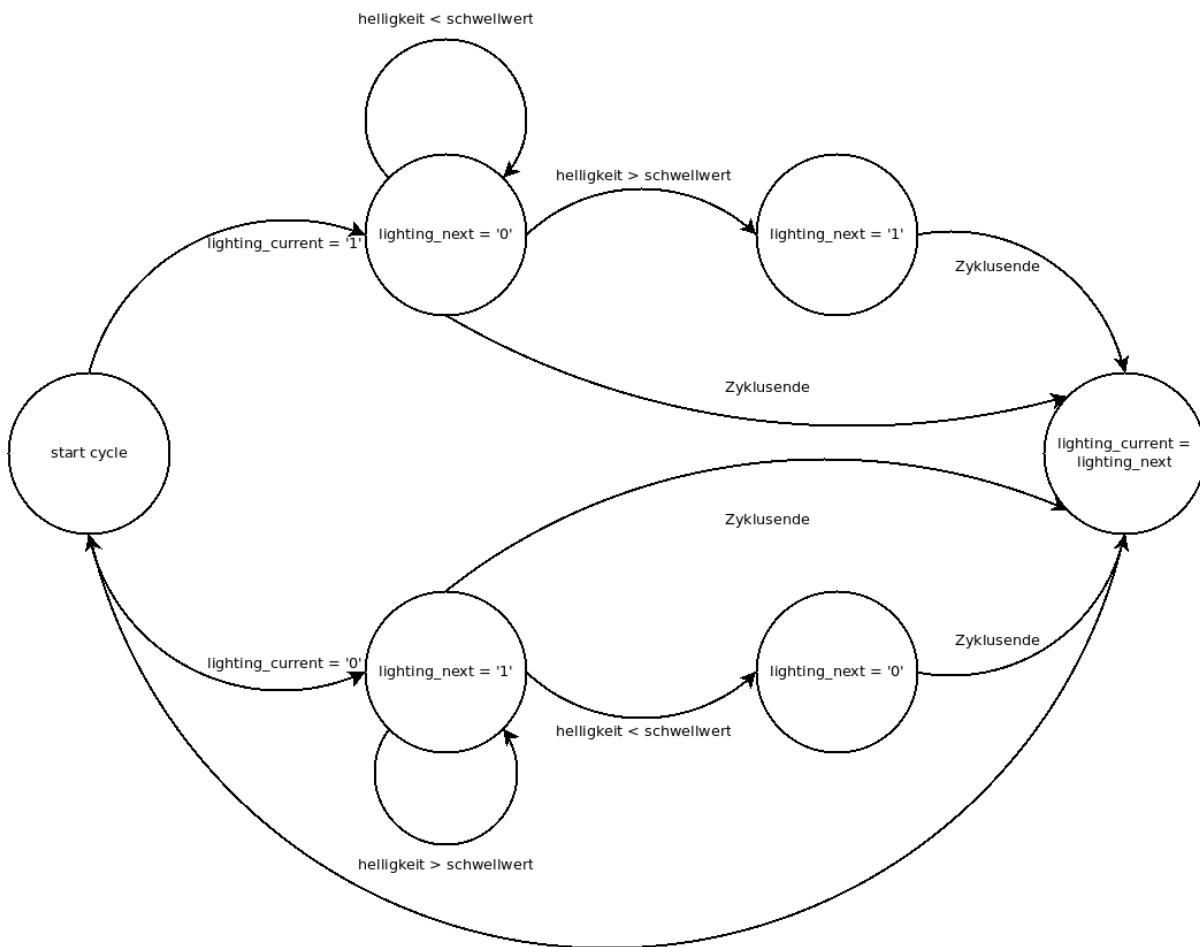
VHDL-Implementierung: `src/vhdl/modules/peripherals.vhdl`

4.2.1 Beleuchtung

Die Beleuchtung umfasst zwei LED-Streifen mit einer Gesamtleistung von 2,25 W, wovon ein Streifen ausschließlich rotes und der andere nur blaues Licht aussendet. Es wurde auf weißes Licht verzichtet, da das in den Pflanzen befindliche Chlorophyll Licht aus dem roten und blauen Wellenlängenbereich benötigt (circa 680 nm und 450 nm).

Insofern der Helligkeitswert unter 400 lx fällt, wird die künstliche Beleuchtung eingeschaltet, um ein gutes Pflanzenwachstum zu garantieren. Damit nicht jede Sekunde das Licht ein- und ausgeschaltet wird, muss im eingeschalteten Zustand, das heißt es wird gerade beleuchtet, der Helligkeitswert innerhalb eines fünf Sekunden langen Zyklus stets über dem Schwellwert von 400 lx sein, damit die Beleuchtung ausgeschaltet wird. Insofern die Beleuchtung sich im ausgeschalteten Zustand befindet, muss der Helligkeitswert innerhalb eines fünf Sekunden langen Zyklus stets unter dem Schwellwert sein, damit die Beleuchtung angeschaltet wird.

Das Zustandsdiagramm verdeutlicht den beschriebenen Ablauf:



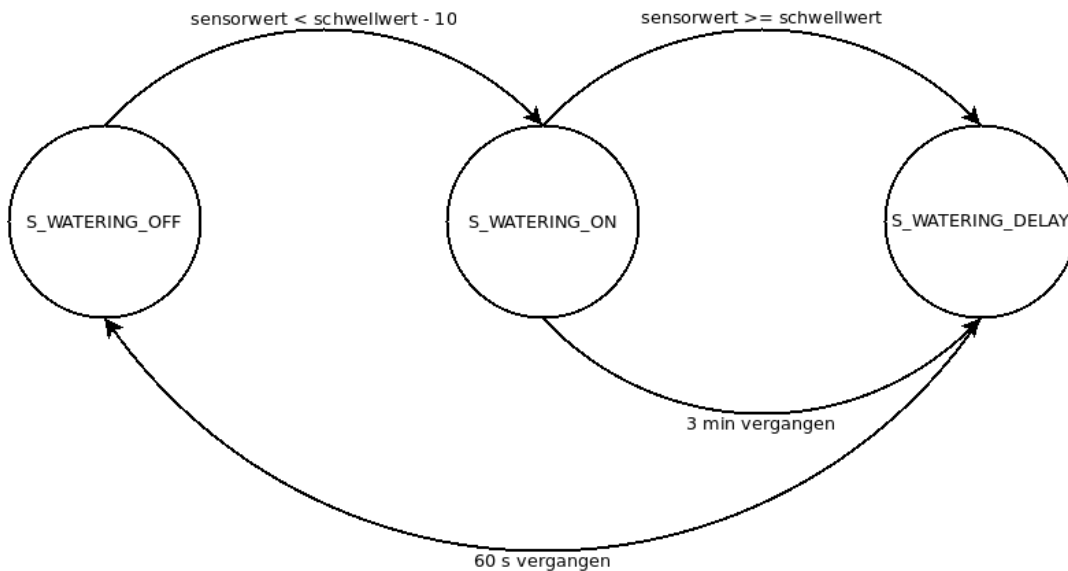
Bei leicht bewölktem Himmel zur Mittagszeit wurde beispielsweise eine Helligkeit von 2100 ~ 2200 lx gemessen. In einem Wohnraum am Tag ohne Zimmerbeleuchtung von 100 ~ 200 lx.

Die Webseite Zimmerpflanzenlexikon schreibt dazu:

„Pflanzen lassen sich in grob drei Kategorien unterteilen: Pflanzen mit hohem Lichtbedarf über 1.000 bis 1.500 Lux, Pflanzen mit mittlerem Lichtbedarf von 800 bis 1.000 Lux und Pflanzen mit geringem Lichtbedarf über 500 Lux. Unter 500 Lux findet keine Photosynthese mehr statt.“
(<http://www.zimmerpflanzenlexikon.info/artikel/lichtbedarf-von-pflanzen/>)

4.2.2 Bewässerung

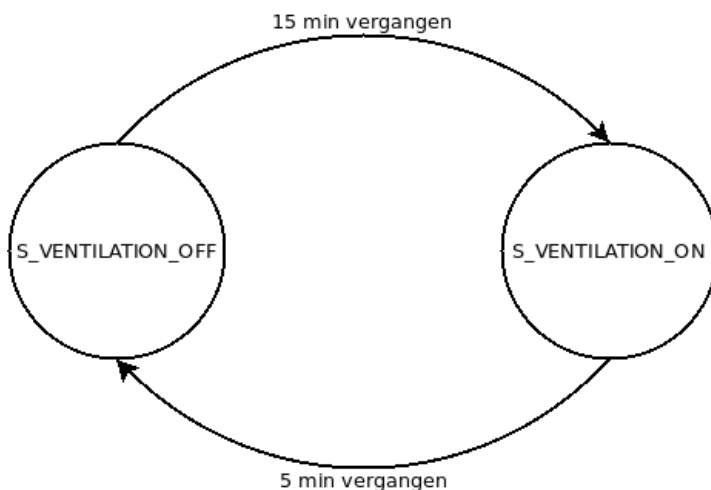
Bei der Steuerung der Bewässerung musste beachtet werden, dass sich das Wasser im Boden erst "verteilen" muss, bevor der Feuchte-Sensor einen aussagekräftigen Wert liefern kann. Deshalb wartet die FSM zur Steuerung der Bewässerungspumpe eine Minute, bevor der Messwert des Feuchte-Sensors berücksichtigt wird. Insofern dieser dann geringer als der gewünschte Schwellwert minus zehn ist, so wird die Bewässerung für maximal drei Minuten angeschaltet. Falls während dieser drei Minuten der Bewässerung der Schwellwert überschritten wird, wechselt die FSM sofort in den ausgeschalteten Zustand `S_WATERING_DELAY`. Als Schwellwert wurde 420 genommen.



Da die Pumpe zwar bei 12 V betrieben werden kann, jedoch manchmal nicht anläuft, wurde die Spannung mithilfe eines Step-Up Converters auf 24 V erhöht. Damit gibt es keine Schwierigkeiten beim Anlaufen der Pumpe und eine zuverlässige Bewässerung ist garantiert.

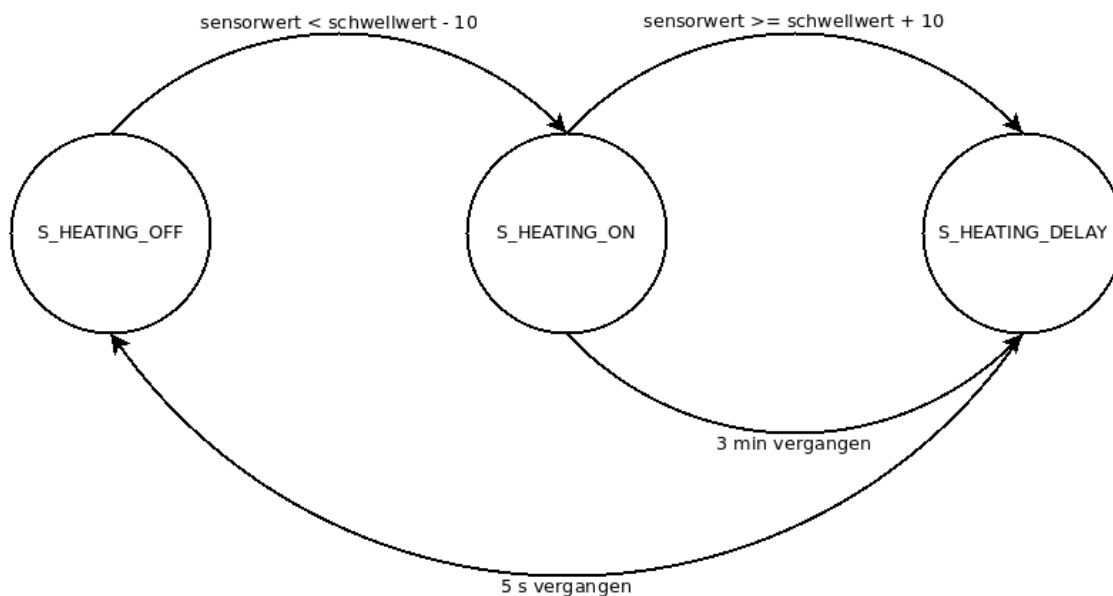
4.2.3 Belüftung

Die Belüftung besteht aus einem Elektromagneten, der ein Fenster öffnet und schließt, und einem kleinen Axial-Lüfter, der den Gasaustausch mit der Außenluft begünstigen soll. Beide Komponenten werden mit einem Relais gesteuert. Da dieser Mechanismus von keiner gemessenen Größe beeinflusst wird, basiert es auf einem zeitlich festgelegten Schließ-/Öffnungsablauf. Das heißt, dass die Belüftung immer alle 15 Minuten für 5 Minuten aktiv ist. Das Zustandsdiagramm ist demnach auch recht simpel:



4.2.4 Heizung

Zwischen den beiden an der Heizplatte liegenden Heizfolien befindet sich der Temperatursensor. Dieser misst die aktuelle Temperatur der Heizplatte. Insofern die gemessene Temperatur mindestens um ein ganzes Grad Celsius niedriger ist als die gewünschte Temperatur, so ändert sich der Zustand der für die Steuerung der Heizung verantwortliche FSM zu `S_HEATING_ON` und es wird geheizt. Der Heizvorgang dauert solange, bis eine Temperatur von mindestens einem Grad Celsius über dem angestrebten Wert erreicht wird oder bereits drei Minuten ununterbrochen geheizt wurde. Dieser "Timeout" nach drei Minuten soll verhindern, dass permanent geheizt wird. Danach wechselt die FSM in den Zustand `S_HEATING_DELAY`, sodass fünf Sekunden gewartet wird, bis wieder der Zustand `S_HEATING_OFF` erreicht wird. Der angestrebte Wert kann im Code verändert werden und liegt standardmäßig bei 26,0 °C



4.3 Ressourcennutzung

Der "Compilation Report" nach der Synthese des VHDL Codes verdeutlicht, dass nur ein Bruchteil der zur Verfügung stehenden logischen Elemente oder des verfügbaren Speichers letztlich genutzt werden. Das bedeutet, dass das Projekt auf einem deutlich kleineren Chip als den des DE2-Boardes umgesetzt werden kann.

Total logic elements	3,625 / 114,480 (3 %)
Total combinational functions	3,499 / 114,480 (3 %)
Dedicated logic registers	1,482 / 114,480 (1 %)
Total registers	1482
Total pins	201 / 529 (38 %)
Total virtual pins	0
Total memory bits	2,944 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	14 / 532 (3 %)
Total PLLs	1 / 4 (25 %)

4.4 Stromverbrauch

Das gesamte Modell verbraucht circa 60 W, was hauptsächlich an dem hohen Verbrauch des Computernetzteiles liegt (24 W). Dieser tritt auf, wenn keine Peripherie eingeschaltet ist, das heißt im "Stand-By". Es ist deshalb für diese Anwendung aufgrund des Verbrauches ungeeignet. In dieser Hinsicht gäbe es weitaus effizientere Netzteile, die man verwenden könnte.

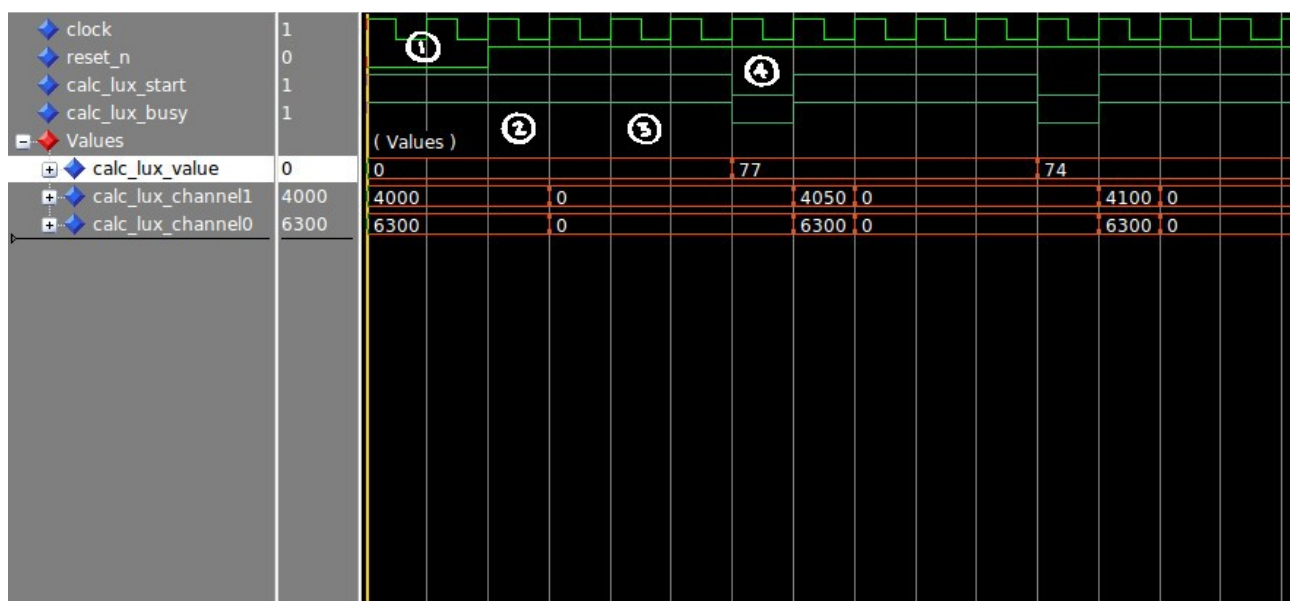
Das DE2-Board inklusive der gesamten Sensorik benötigt gerade einmal 4 W und hat damit einen weitaus besseren und effizienten Verbrauch. Hier eine Liste mit den Verbrauch der restlichen Komponenten.

Aktorik	Stromstärke	Verbrauch
Lüfter und Elektromagnet	200 mA	2,4 W
Heizfolien	2000 mA	24 W
Wasserpumpe	700 mA	8,4 W
Beleuchtung	188 mA	2,25 W

4.5 Testbenches

Insgesamt wurde nur eine Testbench effektiv benötigt, da die Fehler beim Auslesen der Sensorwerte auf äußeren Einflüssen beruhten, das heißt bei einer Simulation nicht auftraten, und die Umsetzung einer I²C Testbench äußerst schwierig erschien. Aber auch bei der Einbindung vom LC-Display und den 7-Segment Anzeigen wurde auf eine Simulation verzichtet, weil diese Umsetzung schnell und ohne Komplikationen funktionierte. Zur Zeitersparnis wurde also weitestgehend auf Testbenches verzichtet außer bei größeren Unsicherheiten, wie bei der Realisierung der Umrechnung der Sensorwerte des Helligkeitssensors.

Um dies zu ermöglichen, wurde eine Funktion erstellt, die den korrekte Helligkeitswert zurückgeben sollte. Aufgrund dessen, dass ein Verhältnis von den beiden Kanalwerte gebildet werden musste, wurde eine Divisions-Funktion hinzugefügt. Zwar war diese Umsetzung in der Simulation funktionstüchtig, allerdings nicht synthetisierbar. Deshalb wurde eine Division von Werten umgangen und aus der Funktion zum Kalkulieren des Lux-Wertes eine Entity gemacht. Diese berechnet den Helligkeitswert über mehrere Taktzyklen verteilt, um einen kritischen Pfad auszuschließen und den Code komplett synthetisierbar zu halten.



- Ablauf:**
- invertiertes Reset Signal (``reset_n``) ist auf logisch '0', um die Entity zu initialisieren (siehe Nr. 1)
 - im zweiten Schritt (siehe Nr. 2) werden die Kanalwerte von der Testbench auf die Werte 6300 und 4000 gesetzt
 - über drei Taktzyklen (siehe Nr. 3) wird Helligkeitswert berechnet (Koeffizienten aus dem Verhältnis von Kanal 1 zu Kanal 0 ermitteln, Kanäle mit Koeffizienten multiplizieren und miteinander subtrahieren, Skalieren des Resultates)
 - Berechnung fertig, ``busy`` Signal geht auf logisch '0' und ``calc_lux_value`` kann ausgelesen werden (siehe Nr. 4)

Damit die Berechnung erneut gestartet werden kann, muss der Entity ein Acknowledgement erhalten, das heißt, dass der berechnete Werte ausgelesen wurde. Dies geschieht, indem das ``start`` Signal auf logisch '0' gesetzt wird (siehe Abbildung, Nr. 4)

Berechnet man den Wert auf die Weise, wie sie auf der Hersteller Website für einen Arduino Uno umgesetzt wurde (https://github.com/Seeed-Studio/Grove_Digital_Light_Sensor), so ergibt sich:

$$x = \frac{322}{81} \cdot channel0 = \frac{322}{81} \cdot 6300 = 25044,44$$

$$y = \frac{322}{81} \cdot channel1 = \frac{322}{81} \cdot 4000 = 15901,23$$

$$ratio = \frac{y}{x} = \frac{15901,23}{25044,44} = 0,635 \Rightarrow b = 0,0128 \quad m = 0,0153$$

$$lux = x \cdot b - y \cdot m = 25044,44 \cdot 0,0128 - 15901,23 \cdot 0,0153 = 77,28$$

Dieser Wert stimmt mit dem berechneten Wert aus der Simulation überein.

5 Schlusswort

Zusammenfassend kann man sagen, dass das Projekt in der Lage ist, eine Gewächshaus zu automatisieren und Pflanzen eine optimale Umgebung zu schaffen im Hinblick auf Wärme, Licht und Feuchte. In wie weit das Wachstum im Vergleich zu „normal“ angebauten Pflanzen aus dem Garten ist, konnte leider nicht aufgrund der mangelnden Zeit getestet werden. Außerdem ist der Stromverbrauch des Netzteiles noch zu hoch, als dass man langfristig das Projekt angeschaltet lassen könnte. Dieses Problem entsteht jedoch aufgrund des eher ungeeigneten Computer-Netzteils und kann problemlos durch effizientere Netzteile behoben werden.

Trotz Einbindung eines CO₂-Sensors, welcher sehr aufwändig und teuer ist, sind die Projektkosten relativ gering ausgefallen, was positiv anzumerken ist. Außerdem ist auch die Ressourcennutzung innerhalb des FPGA äußerst gering, weshalb man bei einer Produktion einen deutlich kleineren Chip als der auf dem DE2-Board verbaute verwenden kann.