

Schlussbericht
Schneller Pathogennachweis mit Resistenztest im Mikroflusssystem - SeptoSurvive
Teilvorhaben: SeptoSurvive – Solectrix GmbH

Zuwendungsempfänger:

Solectrix GmbH

Vorhabenbezeichnung:

Laufzeit des Vorhabens:

Förderkennzeichen:

13GW0431E

SeptoSurvive

01.01.2021 – 31.12.2024

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Inhaltsverzeichnis

| | | |
|-------|--|----|
| 1 | Teil II: Eingehende Darstellung | 4 |
| 1.1 | Verwendung der Zuwendung und erzielte Ergebnisse | 4 |
| 1.1.1 | AP SOL1: Konzept echtzeitfähige Steuerung für den aktiven Sorter | 4 |
| 1.1.2 | AP SOL2: Konzept zur echtzeitfähigen Erregererkennung | 5 |
| 1.1.3 | AP SOL3: Autofokus und Aktorik für phasenfluorimetrische Wachstumsmessung | 15 |
| | AP SOL4: Konzept zur Bestimmung von Wachstumskinetiken aufgrund phasenfluorimetrisch erfasster Sauerstoffverbrauchswerte | 15 |
| 1.1.4 | AP SOL5: Aufbau eines Demonstrators zur Konzeptentwicklung eines Gesamtsystems | 24 |
| 1.1.5 | AP SOL6: Konzept zur Erkennung von antibiotischen Resistenzprofilen aufgrund von Wachstumskinetiken | 25 |
| 2 | Literaturverzeichnis | 26 |

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1 FIRE Echtzeit-Steuerung..... | 4 |
| Abbildung 2: Aufteilung Gesamtaufbau Solectrix / ILT..... | 5 |
| Abbildung 3 Prinzip Bild Fluoreszenz- und Streulichterfassung..... | 5 |
| Abbildung 4: Blockdiagramm Messelektronik..... | 8 |
| Abbildung 5: Anschlussbelegung FIRE-B Board..... | 8 |
| Abbildung 6 Leiterplattenmodell des FADC-Boards zur Integration in das FIRE-B System..... | 9 |
| Abbildung 7: Kommunikationsschema | 9 |
| Abbildung 8 FPGA-Architektur | 11 |
| Abbildung 9: Grafische Oberfläche SeptoSurvive | 13 |
| Abbildung 10: Beispiele aus Embedded Software Applikation..... | 14 |
| Abbildung 11: Beispielaufnahmen zweier unterschiedlicher Bakterientypen ohne Zugabe von Atibiotika. Links: e_coli, rechts: pseudonomas aeruginosa..... | 15 |
| Abbildung 12: Zeitliche aufeinanderfolgende Einzelbilder (oben links: t=16, oben rechts: t=17) für e_coli und zugehöriges Differenzbild (unten)..... | 16 |
| Abbildung 13: Einzelbild für e_coli (t=0, z=40) und zugehörige Binärmaske um Beads und örtlichen Versatz zu maskieren..... | 17 |
| Abbildung 14: Algorithmische Übersicht zur Wachstumsanalyse | 17 |
| Abbildung 15: Beispielhafte .json Konfiguration | 18 |
| Abbildung 16: Featurepunkte für zwei aufeinanderfolgende Zeitpunkte t (oben, links) und t+1 (oben, rechts) und gematchte Featurepunkte für Zeitpunkt t (unten) | 20 |
| Abbildung 17: Minimumbild (oben, links), binäre Maske (oben, rechts), maskiertes Minimumbild (unten) | 21 |
| Abbildung 18: Minimumbild (links) und Segmentierungsergebnis (rechts)..... | 22 |
| Abbildung 19: Wachstumserkennung für e_coli zu den Zeitpunkten t=5 (oben, links), t=10 (oben, rechts), t=15 (unten, links) und t=20 (unten, rechts)..... | 23 |
| Abbildung 20: E_coli: normierte Wachstumskurve (links), sigmoidale Annäherung (Mitte) und relative Wachstumskurve (rechts) | 23 |
| Abbildung 21: E_coli unter Zugabe von Antibiotikum: normierte Wachstumskurve (links), sigmoidale Annäherung (Mitte) und relative Wachstumskurve (rechts)..... | 24 |
| Abbildung 22: Vergleich: OPAL Sauerstoffmesskurve (links) und ermittelte Wachstumskurve (rechts) | 24 |
| Abbildung 23: Demonstratorkonzept mit Teillösungen..... | 25 |
| Abbildung 24: Detektionsergebnisse und Wachstumskurven ohne Antibiotikazugabe (links), mit Antibiotikazugabe (rechts) und verdünnter Antibiotikazugabe (Mitte) | 26 |

1 Teil II: Eingehende Darstellung

1.1 Verwendung der Zuwendung und erzielte Ergebnisse

1.1.1 AP SOL1: Konzept echtzeitfähige Steuerung für den aktiven Sorter

Die Anforderungen an die echtzeitfähige Steuerung für den aktiven Sorter wurden gemeinsam mit dem ILT diskutiert und präzise definiert. Die größten Herausforderungen liegen dabei in den für das Vorhaben abgeleiteten folgenden Systemanforderungen:

- Benötigte Reaktionszeit für die erfolgreiche Sortierung liegt im Bereich bis maximal 3 Mikrosekunden
- Zielstellung ist eine Separationsrate von Mikropartikeln im Bereich von 0,5 bis 5µm

Wie schon im Antrag dargelegt, benötigt das Konzept eine echtzeitfähige Steuerung, die im Wesentlichen auf FPGA-Technologie basiert, um die geforderten Reaktionszeiten sicher zu erfüllen.

Eine bereits am ILT etablierte Steuerungslösung, welche erweitert werden kann, hat sich nach tiefergehender Prüfung als geeignete Basis für den weiteren Projektverlauf herausgestellt. Diese Auswahl eines bestehenden Grundstockes wurde gerne herangezogen, um spätere Projektrisiken reduzieren zu können.



FPGA-Interfaceboard für Echtzeit-Rechnungen » FIRE «

Abbildung 1 FIRE Echtzeit-Steuerung

Für die Erreichung der Ziele von SeptoSurvive wurden jedoch im Vergleich zum Bestehenden eine wesentliche schnellere und präzisere Erfassung von Fluoreszenz und Streulicht benötigt. Hierauf zielte die Weiterentwicklung im Vorhaben folglich ab.

Das Konzept der echtzeitfähigen Steuerung wurde als Kombinationslösung aus Steuerungs-PC, Hard-IP ARM CPU (HPS) mit echtzeitfähigem Linux Betriebssystem, sowie einer kombinierten FPGA-Einheit wurde festgelegt.

Das nachfolgende Blockschaltbild zeigt sehr gut die zu erweiternden und neu zu erstellenden Arbeitspakete für die Erreichung der Vorhabensziele.

Die abschließende Realisierung in Elektronik, Software und PGA wurde zusammenhängend mit AP SOL2 geleistet und eng abgestimmt in den Gesamtaufbau des ILT integriert.

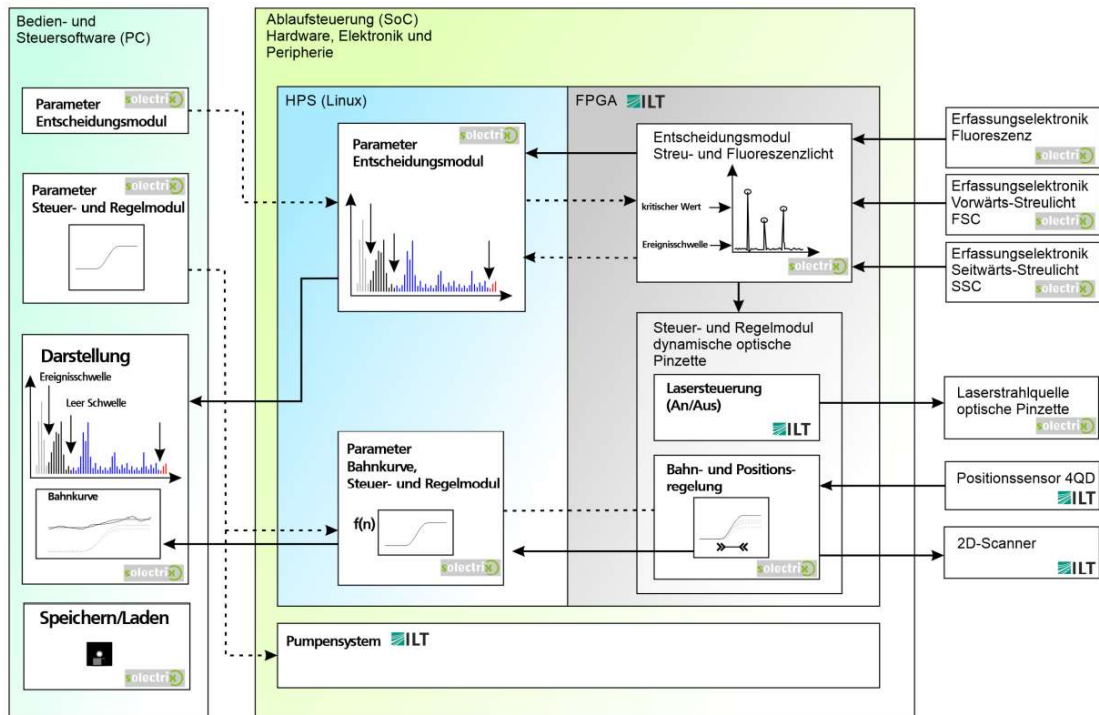


Abbildung 2: Aufteilung Gesamtaufbau Solectrix / ILT

1.1.2 AP SOL2: Konzept zur echtzeitfähigen Erregererkennung

Das Arbeitspaket AP SOL2 definiert die echtzeitfähige Diskriminierung von Erregern/Blutzellen auf Basis der Auswertung von Streulicht. AP SOL2 ergänzt die in AP SOL1 initial mit dem ILT konzeptionierte Steuerungslösung und zielt auf die Realisierung ab.

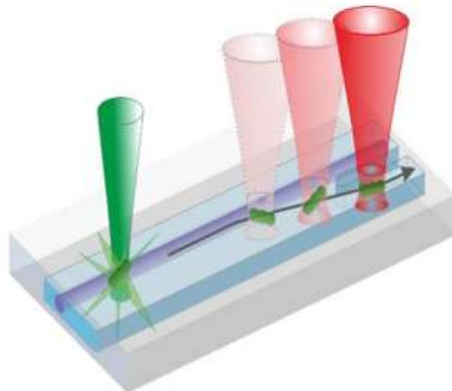


Abbildung 3 Prinzip Bild Fluoreszenz- und Streulichterfassung

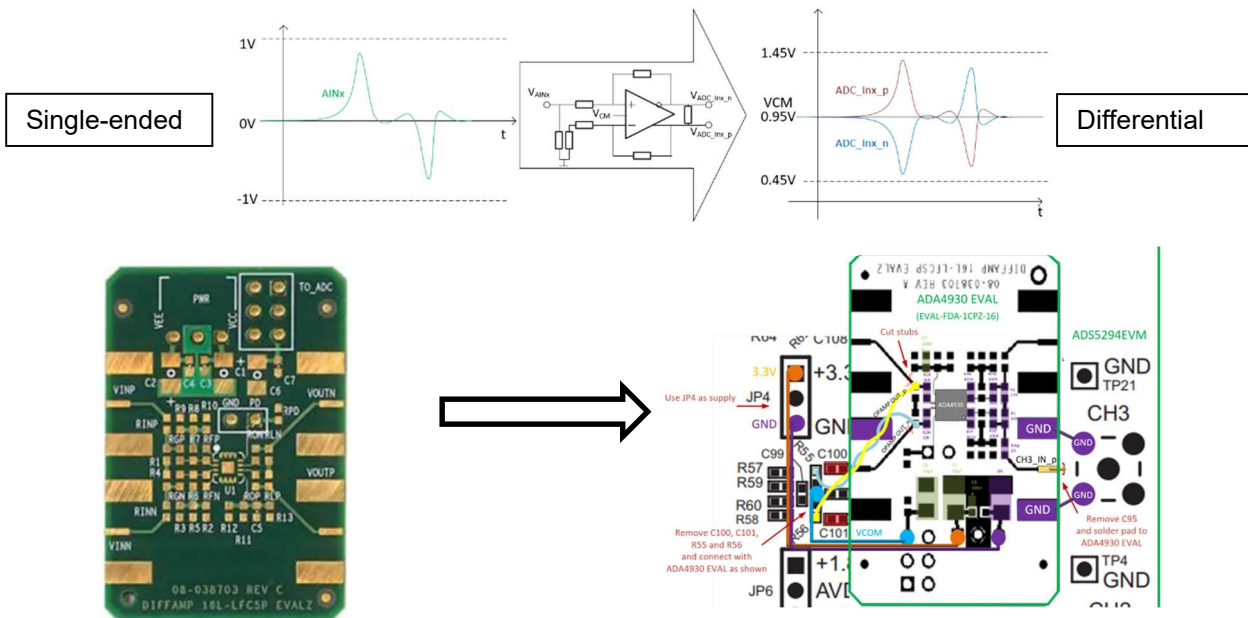
1.1.2.1 Entwicklung Messelektronik

Da die Erfassung von Fluoreszenz- und Streulicht spezielle Anforderungen an die Hardwareerkennungseinheit erfordert und die Realisierbarkeit durch reine, theoretische Datenanalyse nicht möglich war, wurde im ersten Schritt eine Evaluierungslösung entwickelt um das Projektrisiko an dieser Stelle zu minimieren

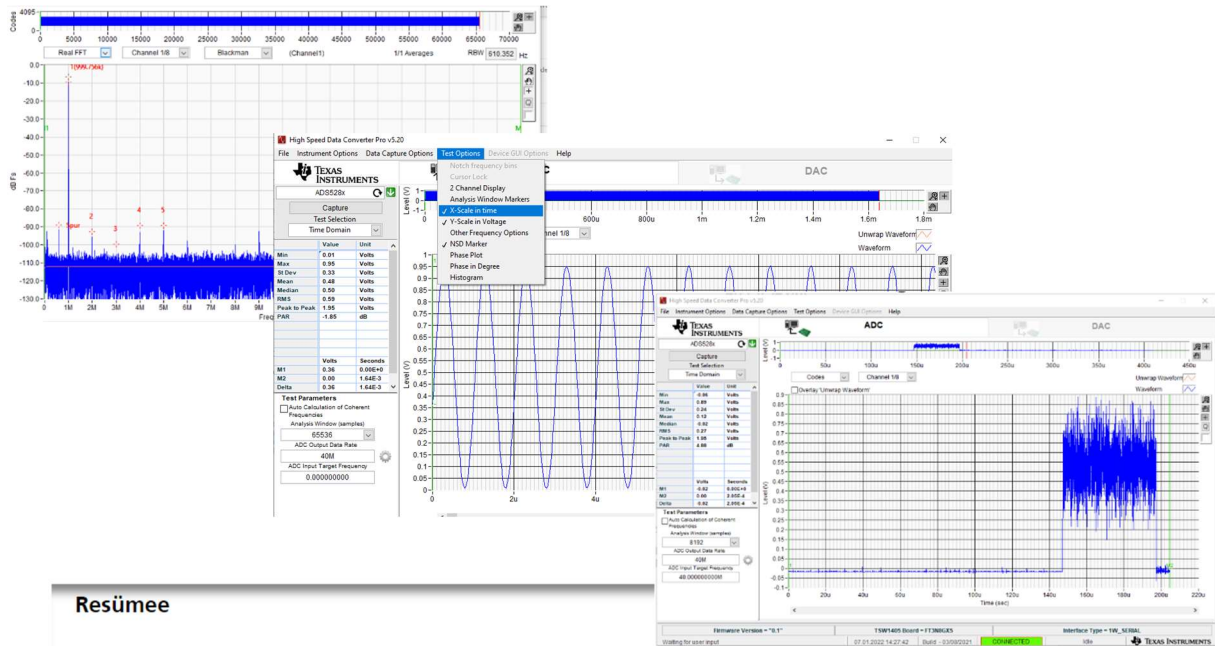
Zur Evaluierung der Analog zu Digitalwandlung wurde ein Aufbau aus einem ADS5294EVM zusammen mit einem TSW1405EVM gewählt:



Für die analoge Eingangsstufe wurde eine OPAMP-Schaltung auf Basis des AD4930 gewählt und durch entsprechende Bestückung und Modifikation einer ADM4930 EVAL Platine an die ADS5294EVM Schaltung angepasst:



Mit Hilfe einer Evaluierungssoftware wurde die Signalstrecke hinsichtlich ihrer Charakteristika vor Implementierung auf Eignung geprüft und bewertet:



Resümee

- AD-Wandler ADS5294 zeigt in ersten Evaluationsversuchen ausreichende Performance
- Performance-Verbesserungen sind durch Anpassen der Tiefpass-Filterung zu erwarten (experimentelle Untersuchung steht noch aus)
- Eine Erhöhung der Messfrequenz könnte die Performance zusätzlich steigern (Aufwand?)
- Bei Verwendung des ADS5294 sollten alle 8 Kanäle realisiert werden, es entfallen dann langsamere AD-Wandler für die Positionsdetektion
- Umsetzung von 14 Bit sollen angestrebt werden

Auf dieser Evaluierungsbasis wurden parallel auch am ILT detaillierte Tests durchgeführt und daraus abgeleitet die Anforderungen an eine konkrete Realisierung in Hard- und Software spezifisch für das Vorhaben finalisiert.

Nach erfolgreicher Evaluierung wurde in die Umsetzung einer spezifischen Elektroniklösung eingestiegen. Als Kernkomponente wurde der über die Evaluierung bestimmte ADC des Typs ADS5294 herangezogen.

Folgende Hauptmerkmale sind relevant:

- 8 Kanäle (je Kanal ein/zwei LVDS diff. pair)
- 14Bit
- 50Mps im 1-wire Betrieb und 80Mps im 2-wire Betrieb



Das nachstehend abgebildete Blockdiagramm zeigt die im nächsten Schritt umzusetzende Elektronik für die hochpräzise Messdatenerfassung:

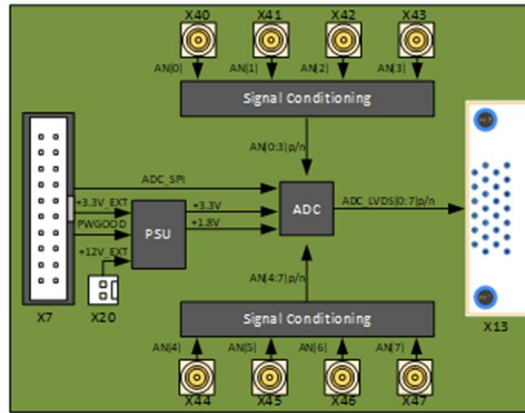
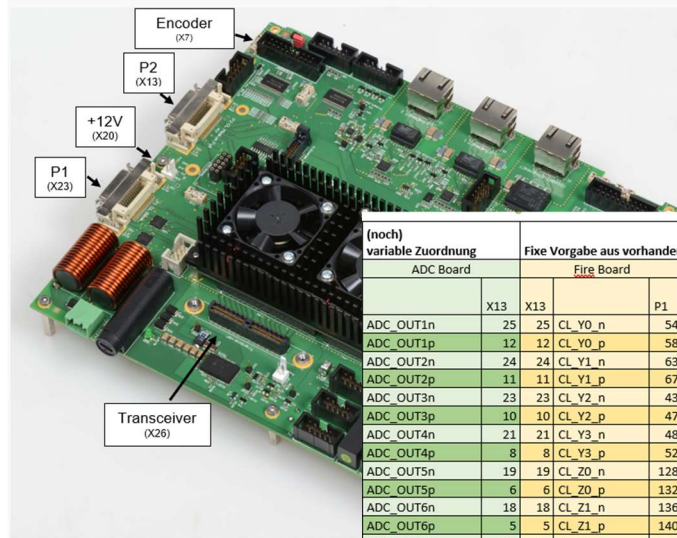


Abbildung 4: Blockdiagramm Messelektronik

Zielstellung war es die Integration dieser spezifischen Elektronik in die vorhandene FIRE-B Umgebung des ILT über die vorhandenen Stecker umsetzen zu können:



| (noch) variable Zuordnung | | Fixe Vorgabe aus vorhandenem Design | | | |
|---------------------------|----|-------------------------------------|-------------|--------------|-----|
| ADC Board | | Fire Board | | DBM_SoC3-A10 | |
| | | Arria10 SX 660 | | | |
| ADC_OUT1n | 25 | 25 | CL_Y0_n | 54 | 54 |
| ADC_OUT1p | 12 | 12 | CL_Y0_p | 58 | 58 |
| ADC_OUT2n | 24 | 24 | CL_Y1_n | 63 | 63 |
| ADC_OUT2p | 11 | 11 | CL_Y1_p | 67 | 67 |
| ADC_OUT3n | 23 | 23 | CL_Y2_n | 43 | 43 |
| ADC_OUT3p | 10 | 10 | CL_Y2_p | 47 | 47 |
| ADC_OUT4n | 21 | 21 | CL_Y3_n | 48 | 48 |
| ADC_OUT4p | 8 | 8 | CL_Y3_p | 52 | 52 |
| ADC_OUT5n | 19 | 19 | CL_Z0_n | 128 | 128 |
| ADC_OUT5p | 6 | 6 | CL_Z0_p | 132 | 132 |
| ADC_OUT6n | 18 | 18 | CL_Z1_n | 136 | 136 |
| ADC_OUT6p | 5 | 5 | CL_Z1_p | 140 | 140 |
| ADC_OUT7n | 17 | 17 | CL_Z2_n | 166 | 166 |
| ADC_OUT7p | 4 | 4 | CL_Z2_p | 170 | 170 |
| ADC_OUT8n | 15 | 15 | CL_Z3_n | 146 | 146 |
| ADC_OUT8p | 2 | 2 | CL_Z3_p | 150 | 150 |
| ADC_LCLKn | 16 | 16 | CL_ZCK_n | 135 | 135 |
| ADC_LCLKp | 3 | 3 | CL_ZCK_p | 139 | 139 |
| ADC_ACLKn | 22 | 22 | CL_YCLK_n | 61 | 61 |
| ADC_ACLKp | 9 | 9 | CL_YCLK_p | 65 | 65 |
| GND | 13 | 13 | (GND)* | | |
| GND | 14 | 14 | (GND)* | | |
| GND | 20 | 20 | 100R to -7 | | |
| GND | 7 | 7 | 100R to -20 | | |
| GND | 26 | 26 | (GND)* | | |

| (noch) variable Zuordnung | | Fixe Vorgabe aus vorhandenem Design | | | |
|---------------------------|----|-------------------------------------|-------------|--------------|-----|
| ADC Board | | Fire Board | | DBM_SoC3-A10 | |
| | | Arria10 SX 660 | | | |
| ADC_OUT1n | 25 | 25 | CL_Y0_n | 54 | 54 |
| ADC_OUT1p | 12 | 12 | CL_Y0_p | 58 | 58 |
| ADC_OUT2n | 24 | 24 | CL_Y1_n | 63 | 63 |
| ADC_OUT2p | 11 | 11 | CL_Y1_p | 67 | 67 |
| ADC_OUT3n | 23 | 23 | CL_Y2_n | 43 | 43 |
| ADC_OUT3p | 10 | 10 | CL_Y2_p | 47 | 47 |
| ADC_OUT4n | 21 | 21 | CL_Y3_n | 48 | 48 |
| ADC_OUT4p | 8 | 8 | CL_Y3_p | 52 | 52 |
| ADC_OUT5n | 19 | 19 | CL_Z0_n | 128 | 128 |
| ADC_OUT5p | 6 | 6 | CL_Z0_p | 132 | 132 |
| ADC_OUT6n | 18 | 18 | CL_Z1_n | 136 | 136 |
| ADC_OUT6p | 5 | 5 | CL_Z1_p | 140 | 140 |
| ADC_OUT7n | 17 | 17 | CL_Z2_n | 166 | 166 |
| ADC_OUT7p | 4 | 4 | CL_Z2_p | 170 | 170 |
| ADC_OUT8n | 15 | 15 | CL_Z3_n | 146 | 146 |
| ADC_OUT8p | 2 | 2 | CL_Z3_p | 150 | 150 |
| ADC_LCLKn | 16 | 16 | CL_ZCK_n | 135 | 135 |
| ADC_LCLKp | 3 | 3 | CL_ZCK_p | 139 | 139 |
| ADC_ACLKn | 22 | 22 | CL_YCLK_n | 61 | 61 |
| ADC_ACLKp | 9 | 9 | CL_YCLK_p | 65 | 65 |
| GND | 13 | 13 | (GND)* | | |
| GND | 14 | 14 | (GND)* | | |
| GND | 20 | 20 | 100R to -7 | | |
| GND | 7 | 7 | 100R to -20 | | |
| GND | 26 | 26 | (GND)* | | |

| (noch) variable Zuordnung | | Fixe Vorgabe aus vorhandenem Design | | | |
|---------------------------|-----|-------------------------------------|------------|------------------|----|
| ADC Board | | Fire Board | | DBM_SoC3-A10 | |
| | | Arria10 SX 660 | | | |
| ADC Board | | Fire Board | | Pegelwandler | |
| | X13 | X7 | | U7 (3.3V<->1.8V) | P1 |
| GND | 1 | 1 | GND | | |
| +3.3V | 2 | 2 | +3.3V | | |
| SCLK | 3 | 3 | Encoder1_A | Pin 2 <- Pin 47 | 10 |
| SDATA | 4 | 4 | Encoder1_B | Pin 3 <- Pin 46 | 8 |
| CS_n | 5 | 5 | Encoder1_N | Pin 5 <- Pin 44 | 14 |
| RESET_n | 6 | 6 | Encoder2_A | Pin 6 <- Pin 43 | 12 |
| Powerdown | 7 | 7 | Encoder2_B | Pin 8 <- Pin 41 | 15 |
| Spare_1 | 8 | 8 | Encoder2_N | Pin 9 <- Pin 40 | 16 |
| Spare_2 | 9 | 9 | Encoder3_A | Pin 11 <- Pin 38 | 19 |
| Spare_3 | 10 | 10 | Encoder3_B | Pin 12 <- Pin 37 | 20 |
| Spare_4 | 11 | 11 | Encoder3_N | Pin 13 <- Pin 36 | 23 |
| Spare_5 | 12 | 12 | Encoder4_A | Pin 14 <- Pin 35 | 28 |
| Spare_6 | 13 | 13 | Encoder4_B | Pin 16 <- Pin 33 | 27 |
| Spare_7 | 14 | 14 | Encoder4_N | Pin 17 <- Pin 32 | 32 |
| Spare_8 | 15 | 15 | Encoder5_A | Pin 19 <- Pin 30 | 35 |
| Spare_9 | 16 | 16 | Encoder5_B | Pin 20 <- Pin 29 | 36 |
| Spare_10 | 17 | 17 | Encoder5_N | Pin 22 <- Pin 27 | 39 |
| Spare_11 | 18 | 18 | Encoder_N | Pin 23 <- Pin 26 | 40 |
| GND | 19 | 19 | GND | | |

Abbildung 5: Anschlussbelegung FIRE-B Board

Nach Erstellung und Freigabe der Schaltung und auch des Platinenlayout bei Solectrix wurde das entwickelte Leiterplattenmodell des FADC- Boards in die Produktion überführt.

Nach erfolgter Produktion wurde die Baugruppe bei Solectrix in Betrieb genommen und geprüft, damit die nachgelagerte Anbindung und Realisierung via Software / FPGA zuverlässig arbeiten konnte.

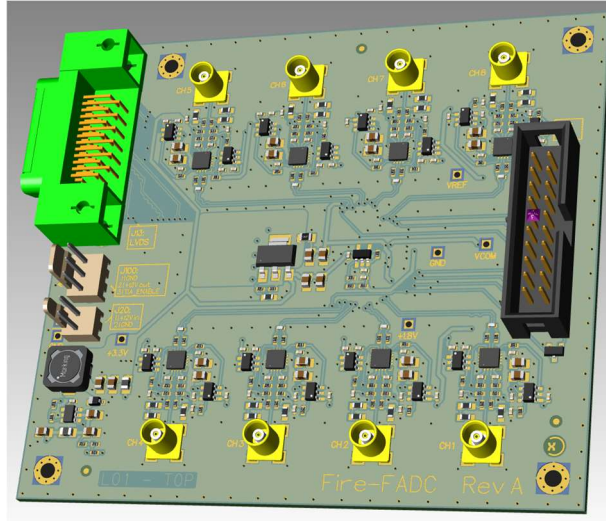


Abbildung 6 Leiterplattenmodell des FADC-Boards zur Integration in das FIRE-B System

1.1.2.2 Verteiltes System / Kommunikation

Für einen zusammenhängenden Betrieb der einzelnen Subsysteme der echtzeitfähigen Erregererkennung wurde ein tragfähiges Grundkonzept entwickelt und umgesetzt. Die daran beteiligten Komponenten sowie die schematische Verknüpfung dieser ist nachfolgend dargestellt.

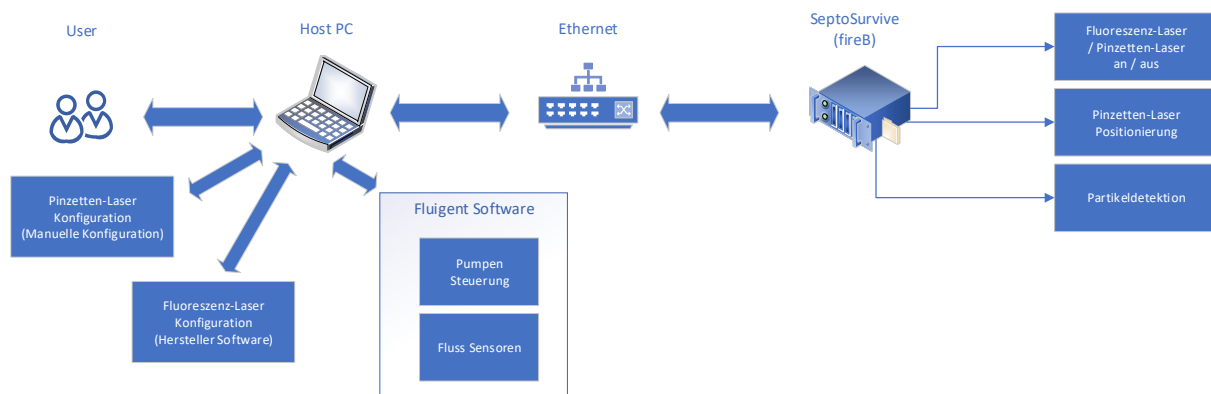


Abbildung 7: Kommunikationsschema

Grundgedanke ist, dass ein verbundener Host PC mittels Ethernetverbindung an die SeptoSurvive Elektronik für die Erregererkennung angebunden ist und diese steuern und überwachen kann.

Auf Seiten des Host PC wird eine grafische Benutzeroberfläche bereitgestellt, welches eine Konfiguration des Messsystems ermöglicht sowie die aufgezeichneten Daten für den Anwender visualisieren kann. Die Kommunikation der PC-Applikation sowie der Komponenten auf Geräteseite

wurde mittels „Google Protocol Buffers“ (protobuf) realisiert, um hier für künftige Weiterentwicklungen robust und skalierbar ausgelegt zu sein.

1.1.2.3 FPGA Funktionalität

Das initiale Grobkonzept aus AP SOL1 wurde verfeinert und dabei im Fokus die Kernfunktionalität des FPGA näher definiert, um damit wie benötigt die gesuchten Partikel detektieren und separieren zu können. Die folgenden Kernaufgaben sollen vom FPGA gesteuert werden:

1. Das FPGA soll 3x Laser (Fluoreszenz-Laser) ansteuern, um die Fluoreszenz der fließenden Partikel anzuregen. Diese Laser werden auf einen Fixpunkt fokussiert.
2. Das emittierte Fluoreszenzlicht soll von den 2x Streulichtsensoren (PDTIA) und von 3x Foto-Multiplier (PMT) gemessen werden. Die Daten werden über einen externen ADC (FireFADC-01 Board) an das FPGA weitergeleitet. Das FPGA wertet diese Informationen aus, um eine Entscheidung zu treffen, ob es sich bei den Messdaten um ein gesuchtes Partikel handelt. Dies muss in Echtzeit geschehen, damit der Pinzetten Laser angesteuert und dieses Partikel noch rechtzeitig einfangen kann.
3. Jede Detektion soll dabei aufgezeichnet werden. Es sollen nur die Daten aufgezeichnet werden, die im zeitlichen Rahmen der Messung zur Auswertung genutzt wurden. (Eventaufzeichnung)
4. Es soll auch möglich sein, alle abgetasteten Sensordaten mit einer Oszilloskop-ähnlichen Funktion auszulesen. Es darf jeweils nur die Oszilloskop- oder die Eventfunktionalität aktiv sein.
5. Das FPGA soll einen Pinzetten Laser (Pinzetten-Laser) ansteuern, um ein detektiertes Partikel einzufangen und aus der Strömung zu trennen. Daher soll die Position vom FPGA unter Verwendung des Acousto Optical Deflector (AOD) gesteuert werden. Die aktuelle Position soll mit der „4QD XY Positionselektronik“ gemessen und geregelt werden. Der Laser folgt einem vordefinierten Pfad, der über Software konfigurierbar ist.
6. Das FPGA soll die Ausgabe der Laser und ihre Leistung steuern
7. Ein Supervisor (System Supervisor) soll integriert werden, um sicherzustellen, dass keine Komponente des Systems durch Überhitzung beschädigt werden kann.
8. Die Konfiguration und die Auswertung der aufgezeichneten Daten erfolgt durch den internen ARM Cortex A9 Prozessor. Die Daten können dabei weiter über Ethernet versendet werden.

In nachfolgender Grafik ist die Umsetzung dieser geforderten Teilmodule innerhalb der FPGA-Implementierung bildlich dargestellt:

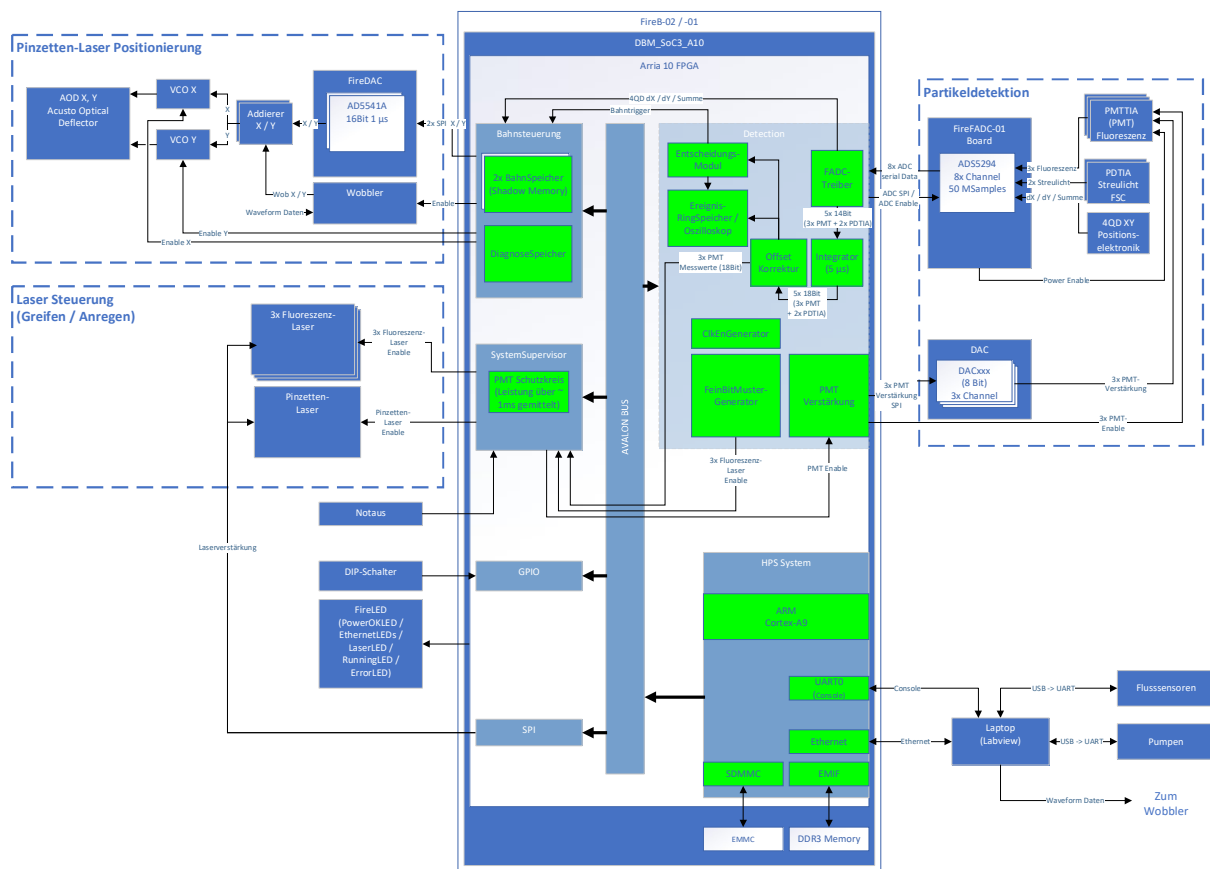


Abbildung 8 FPGA-Architektur

Die wichtigsten Funktionen / Module aus obiger Darstellung sind nachfolgend beschrieben

Detektion

FADC Treiber

Das FADC-Treibermodul erfasst und parallelisiert alle ankommenden seriellen Datenströme für jeden Kanal des ADC Boards. Die parallelisierten Datenwörter entsprechen dabei je einem Sample des jeweiligen Kanals des ADCs und werden an einer einheitlichen Schnittstelle bereitgestellt.

Das Modul sorgt für die Synchronisation der eingehenden Datenströme, indem es feste Testmuster des ADCs konfiguriert. Die SERDES Kanäle synchronisieren sich, auf die eingehenden Daten unter Verwendung des Dynamic Phase Alignment Features der internen Hard-IPs des FPGAs, ein. Das Modul soll eine AVALON-Slave-Schnittstelle bereitstellen, über die der aktuelle Linkstatus sowie alle ADC-internen Register ausgelesen werden können. Außerdem ist es möglich, die ADC-internen Register über die AVALON-Slave-Schnittstelle zu konfigurieren und auszulesen.

Integrator

Ein Integrator-Modul summiert die gesampelten Daten über eine einstellbare Zeit auf. Die Summe der integrierten Daten werden über eine rechts shift Operation (Division von 2) so weit geteilt, damit ein 18 Bit Wert für jeden Kanal erzeugt wird. Somit werden Daten Ausreißer geglättet und man erhält gleichzeitig eine geringere Datenmenge zur Verarbeitung.

OffsetCorrection

Das Modul OffsetCorrection gleicht einen konstanten Offset Anteil der Sensoren aus. Dabei wird das Minimum eines jeden Sensors in Abhängigkeit des aktiven Lasers bestimmt und dann subtrahiert, wenn wiederum dieser Laser aktiv ist. Die Korrektur soll über Software ausschaltbar und das gemessene Minimum auch wieder zurücksetzbar sein.

EreignisRingSpeicher / Oszilloskop

Die aufbereiteten Daten werden in dem Modul EreignisRingSpeicher / Oszilloskop gespeichert. Die Speicherung kann dabei kontinuierlich erfolgen durch die Aufzeichnung sämtlicher Samples oder aber auch event-basierend, sodass nur die Messfenster aufgezeichnet werden, die durch das Entscheidungsmodul festgehalten wurden (Partikeldetektion). Die Umstellung soll dabei über eine Softwareapplikation steuerbar sein.

EntscheidungsModul

Das EntscheidungsModul bestimmt anhand der Charakteristiken der eingehenden Daten der Sensoren und durch die dabei genutzten Laser Einstellung, ob eines der gesuchten Partikeln gefunden wurde. Ein Trigger Signal zeigt den Fund dabei an. Zunächst wird dabei das lokale Maximum und die Fläche unter der eingelesenen Kurve innerhalb eines konfigurierbaren Messfensters bestimmt. Das Messfenster wird dann ausgewertet, wenn es in der Mitte eines lokalen Maximums liegt. Zudem ist es möglich Sensoren zu kombinieren, um ein Trigger nur dann zu erzeugen, wenn mehrere Bedingungen zutreffen, auch unter der Berücksichtigung des zur Messung aktiv gesetzten Lasers.

FeinBitMusterGenerator

Der FeinBitMusterGenerator stellt einen Speicher bereit in dem ein Muster abgelegt werden kann anhand dessen die Laser aktiviert werden. Jeder Eintrag in dem Speicher entspricht dabei 100ns. Dabei ist die Applikationssoftware verantwortlich die An- und Ausschaltzeiten der Laser mit zu berücksichtigen, sowie auch, dass die eingestellten Muster mit der Integrationszeit übereinstimmen.

Bahnsteuerung

Das Modul Bahnsteuerung ist für die Positionierung des Pinzetten Lasers zuständig.

BahnSpeicher

Im BahnSpeicher wird der abzufahren Pfad des Lasers in X- Y-Koordinaten abgespeichert.

DiagnoseSpeicher

Der DiagnoseSpeicher speichert die aktuelle Position des Lasers und kann dazu genutzt werden, um die Laser Positionierung zu regeln.

SystemSupervisor

Der Supervisor stellt sicher, dass keine Komponente des Systems durch Überhitzung beschädigt werden kann. Der aktuelle Status soll über die Applikation Software auslesbar sein. Bei einer automatischen Abschaltung von Laser und PMT muss die Software diese wieder aktiv freischalten.

1.1.2.4 Grafische Bedienoberfläche

Für die Visualisierung der Daten und der Konfiguration der der SeptoSurvive Elektronik wurde eine grafische Bedienoberfläche entwickelt. Diese wurde in C++ unter Verwendung des Qt-Frameworks plattformunabhängig entwickelt, so dass eine Erweiterung der Funktion sowie die Pflege nachhaltig gegeben ist.

Folgende Aspekte wurden im Wesentlichen erreicht:

- Die „SeptoSurvive Applikation“ ist lauffähig unter Windows und Linux
- Die Applikation ist auf Mehrsprachigkeit ausgelegt
- Die Applikation bietet Anwendern eine vereinfachte Benutzereingabe
- Konfigurationen sind im JSON-Format speicher- und ladbar
- Es wird eine Oszilloskopmodus zur Visualisierung erfasster Daten bereitgestellt
- Eine persistente Datenaufzeichnung ist möglich
- Es können alle 5 Messkanäle erfasst werden

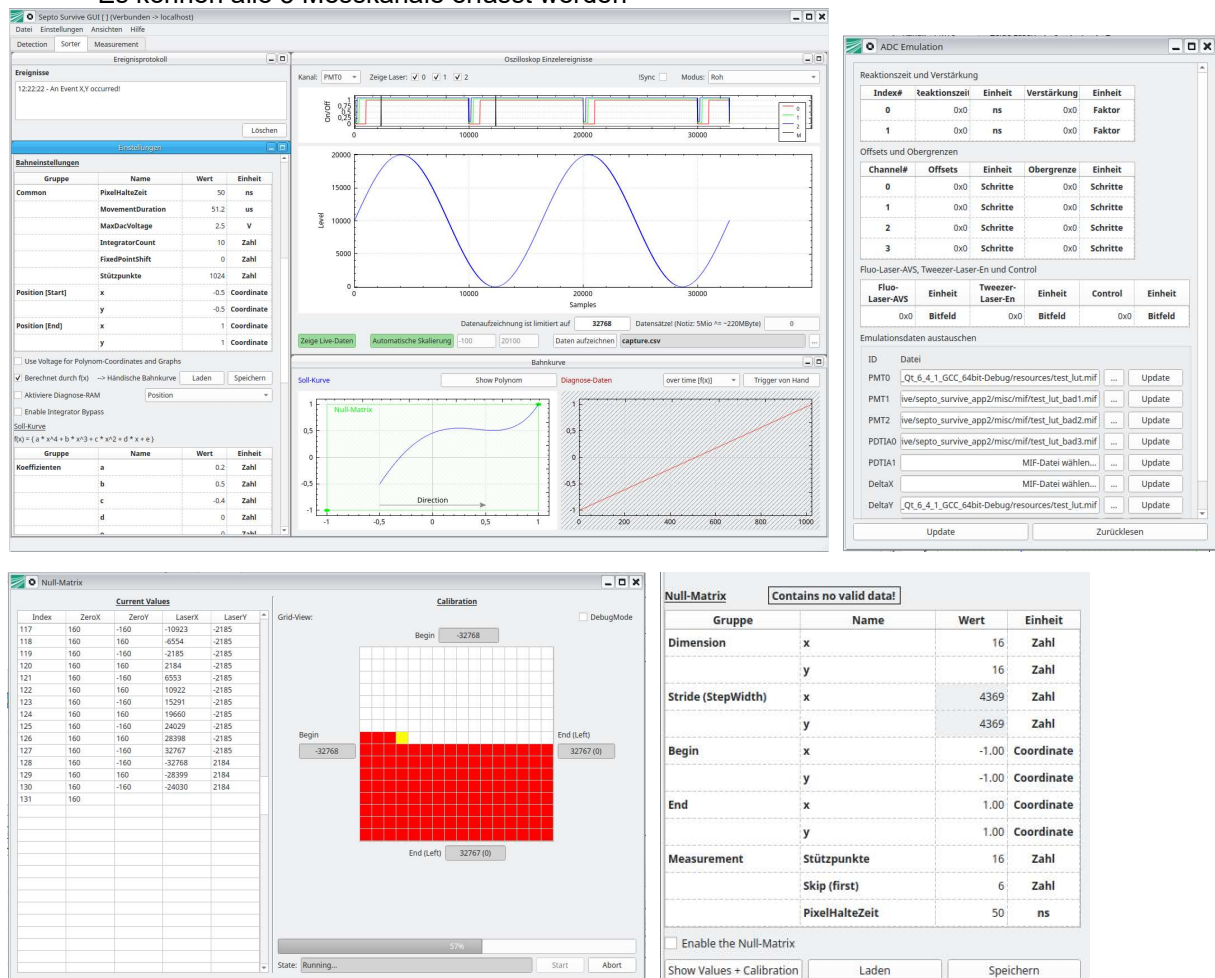


Abbildung 9: Grafische Oberfläche SeptoSurvive

1.1.3 AP SOL3: Autofokus und Aktorik für phasenfluorimetrische Wachstumsmessung AP SOL4: Konzept zur Bestimmung von Wachstumskinetiken aufgrund phasenfluorimetrisch erfasster Sauerstoffverbrauchswerte

Die Arbeiten an AP SOL3 und AP SOL4 wurden während des Vorhabens gemeinsam betrachtet und auch so bearbeitet. Neben der initial geplanten phasenfluorimetrischen Wachstumsmessung wurde in Kooperation mit dem UKA als weiterer Ansatzpunkt auch eine Betrachtung der Wachstumskinetiken anhand von Bilddaten über einen Mikroskopieaufbau begonnen.

Diese ergänzende Variante unter Verwendung eines Mikroskops zur Erzeugung der Bilddaten wird auch als Referenz zur Verifizierung der Messergebnisse der phasenfluorimetrischen Wachstumsmessung (OPAL-System) dienen können.

Im Labor des UKA wurden im Projektzeitraum hiermit fortlaufend Versuchsreihen mit Bakterienstämmen durchgeführt, welche jeweils mittels Bildaufnahmen überwacht und dokumentiert wurden. Die dabei entstandenen Z-Stapelaufnahmen wurden durch *Solectrix* analysiert und bewertet.

1.1.3.1 Datenmaterial

Als Ausgangssituation standen 4D-Bilddaten von zu evaluierenden Bakterienkulturen zur Verfügung. Die Daten konnten dabei in zwei räumlichen Ortsdimensionen (x,y), einer Tiefenrichtung (z) und einer zeitlichen Komponente (t) analysiert werden.

Bildmaterial wurde zu folgenden Bakterientypen erhoben:

- *E.coli*
- *Enterococcus faecium*
- *Pseudomonas aeruginosa*
- *Staph aureus*
- *Vibrio natriegens*

Die Bakterientypen unterschieden sich dabei sowohl in der Geschwindigkeit als auch in der Form des Wachstums. Des Weiteren wurden unterschiedliche Untersuchungen bezüglich Objektivwahl, Zusammensetzung des Nährmediums und/oder der Zugabe von Antibiotika durchgeführt. Ziel war es jeweils, die zugrundeliegende Wachstumskurve mithilfe geeigneter Bildverarbeitungsalgorithmik abzuschätzen.

Die nachfolgende Abbildung 11 zeigt Beispielaufnahmen der Tiefenschicht $z=80$ zum Aufnahmezeitpunkt $t=15$ für zwei unterschiedliche Bakterientypen, jeweils ohne Zugabe von Antibiotika.

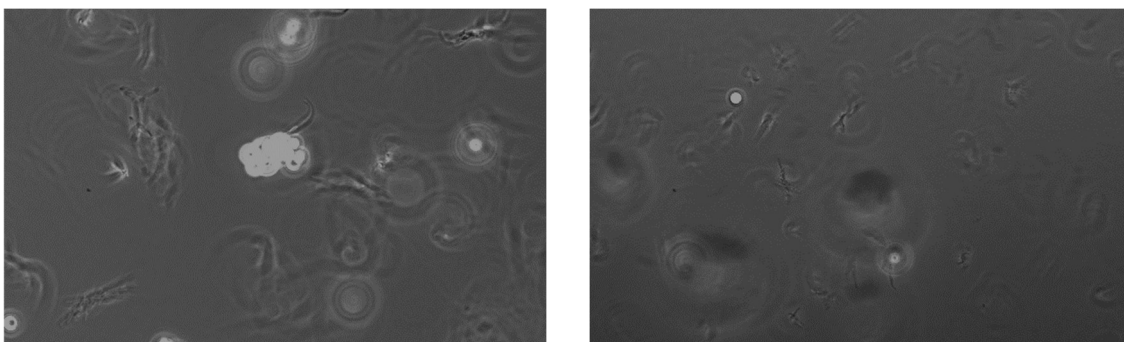


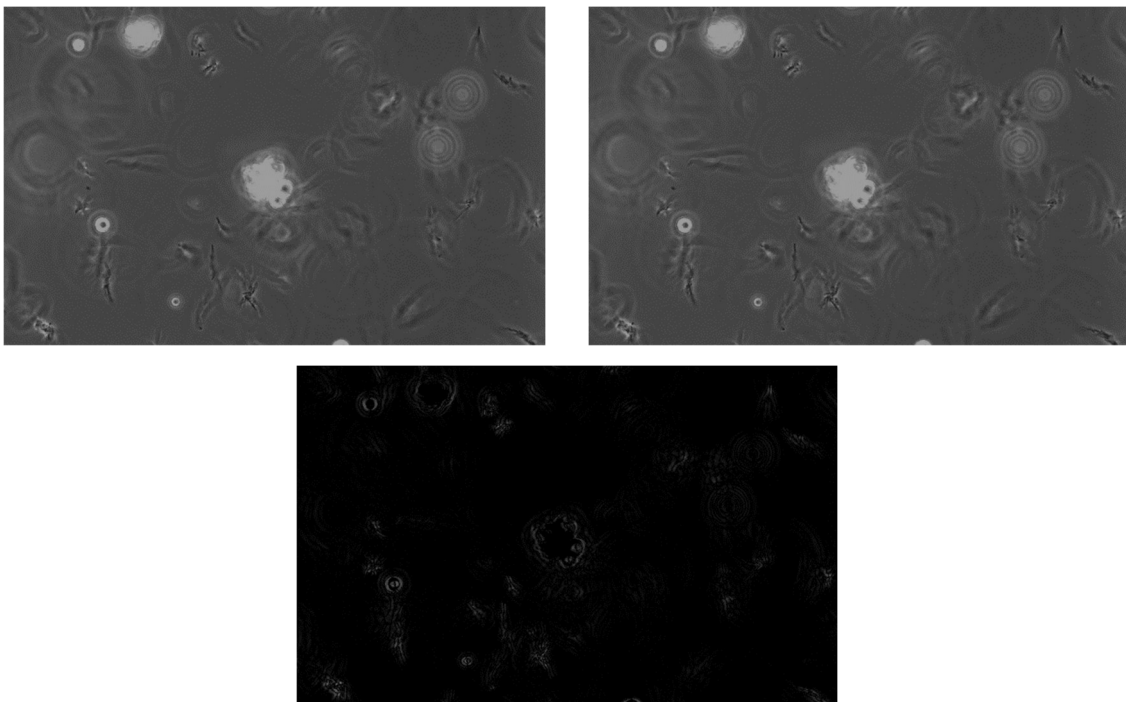
Abbildung 11: Beispielaufnahmen zweier unterschiedlicher Bakterientypen ohne Zugabe von Antibiotika. Links: *e_coli*, rechts: *pseudomonas aeruginosa*

1.1.3.2 Aufnahmebedingte Herausforderungen für die Algorithmik

Dieses Kapitel beschreibt verschiedene algorithmische Herausforderungen die über die eigentliche Objekterkennung hinausgehen. Die Erweiterungen waren aufgrund spezieller Aufnahmeumstände notwendig, um eine zielgerichtete Wachstumsanalyse gewährleisten zu können.

1.1.3.1.1 Örtliche Versatzkorrektur

Die nachfolgende Abbildung 12 zeigt zwei zeitlich aufeinanderfolgende Bilder ($t=16$ und $t=17$) derselben Tiefenschicht ($z=40$) und das zugehörige Differenzbild. Das Differenzbild weist starke Abweichungen zwischen den beiden Bildern auf, die nicht nur wegen des Wachstums sondern vor allem auch aufgrund eines globalen örtlichen Versatzes zwischen den beiden Bildern auftreten. Da sich die Bildverschiebungen zwischen je zwei Aufnahmezeitpunkten über den gesamten zeitlichen Evaluierungsbereich aufsummieren, muss dieser Versatz algorithmisch robust erkannt und korrigiert werden. Dies wird im Unterkapitel 1.1.3.1.5 näher beschrieben. Durch die Versatzkorrektur wird der Analysebereich auf diejenigen Bildbereiche eingeschränkt, die über den gesamten Zeitraum sichtbar sind.



*Abbildung 12: Zeitliche aufeinanderfolgende Einzelbilder (oben links: $t=16$, oben rechts: $t=17$) für *e_coli* und zugehöriges Differenzbild (unten)*

1.1.3.1.2 Maskierung ungültiger Bildbereiche

Um die Sauerstoffkonzentration parallel zur optischen Analyse zu messen, wurden den jeweiligen Proben *Beads* zugegeben. Neben den wachsenden Bakterienkulturen sind damit aber natürlich auch die *Beads* in den einzelnen Schichtbildern sichtbar. Da ein Erkennen dieser Strukturen das Ergebnis der optischen Wachstumsanalyse verfälschen würde, müssen die *Beads* zuverlässig erkannt und maskiert werden. Dies wird ebenfalls in Unterkapitel 1.1.3.1.5 beschrieben.

Die nachfolgende Abbildung 13 zeigt ein Einzelbild für *e_coli* ($t=0, z=40$) und die zugehörige Binärmaske um sowohl die *Beads* Strukturen, als auch den örtlichen Versatz über den zeitlichen Analysebereich zu maskieren.

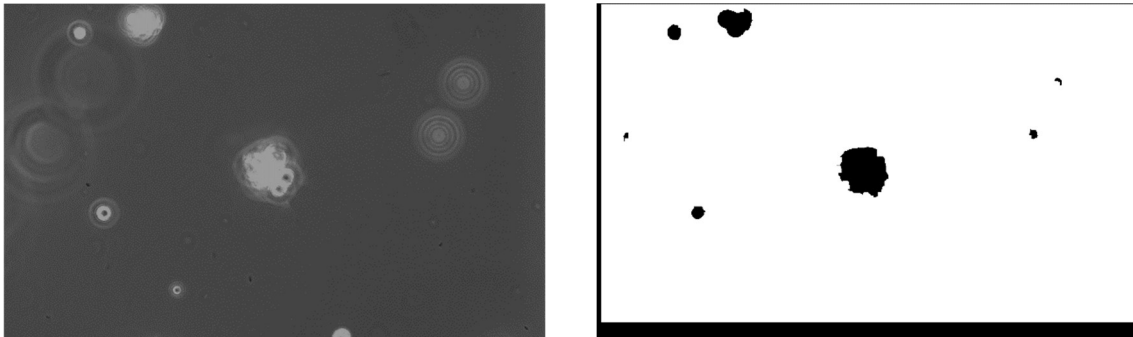


Abbildung 13: Einzelbild für *e_coli* ($t=0, z=40$) und zugehörige Binärmaske um *Beads* und örtlichen Versatz zu maskieren.

1.1.3.3 Algorithmik

Dieses Kapitel beschreibt das algorithmische Vorgehen zur Ermittlung des zu Grunde liegenden Wachstumsverhaltens mit bildbasierten Methoden.

1.1.3.1.3 Übersicht

Die folgende Abbildung 14 zeigt eine Übersicht über die entwickelte Methodik zur Wachstumsanalyse. Die einzelnen Komponenten werden in den nachfolgenden Unterkapiteln 1.1.3.1.4 bis 1.1.3.1.6 näher beschrieben.

Der Image Handler bildet zusammen mit dem Konfigurationsblock die Steuerung der eigentlichen Algorithmik, die mit den Blöcken Versatzkorrektur, Maskierung und Objekterkennung beschrieben wird. Der Results Handler kümmert sich anschließend um eine kompakte Aufarbeitung der Wachstumsergebnisse. Als Programmiersprache wurde Python ausgewählt.

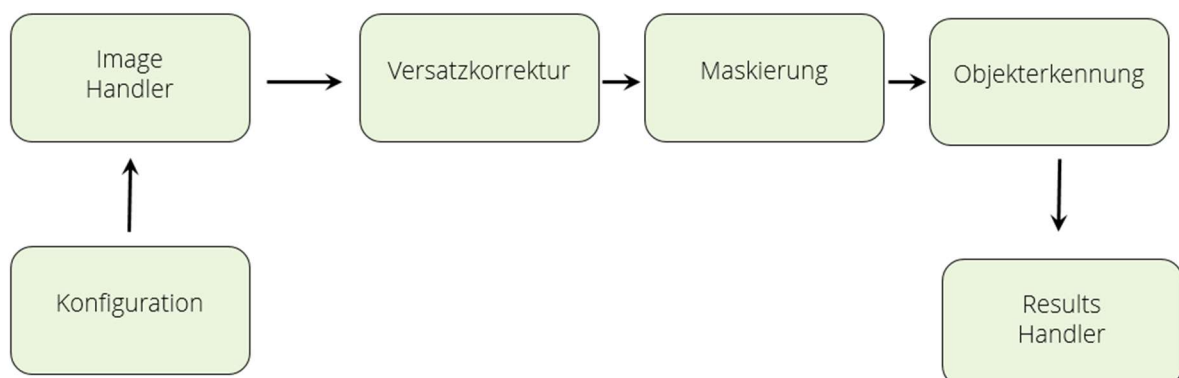


Abbildung 14: Algorithmische Übersicht zur Wachstumsanalyse

1.1.3.1.4 Konfiguration & Image Handler

Die Kernaufgabe des Image Handlers ist das Einlesen der benötigten Einzelbilder aus dem *.vsf* Container-Format, in dem die 4D-Datensätze vorliegen. Dazu wurde die externe Python-Bibliothek *bioformats* verwendet [1]. Durch Angabe der gewünschten Zeit- und Tiefenindizes erlaubt die Bibliothek das Extrahieren beliebiger Bilder aus dem *.vsf* Container.

Das Konfigurationsmanagement wurde mit *.json* Dateien umgesetzt, die eine strukturierte Aufteilung der Konfigurationsparameter auf die einzelnen algorithmischen Blöcke erlauben.

Die nachfolgende Abbildung 15 zeigt eine exemplarische Konfiguration. Für den Image Handler *imgHandler* werden die Top-Level-Parameter, wie Dateipfad, Bildgröße und der zu evaluierende Zeit- bzw Tiefenbereich festgelegt. Für den Results Handler muss unter *resultsHandler* lediglich der Zielpfad angegeben werden. Der Block *imageRegistrator* umfasst Parameter für die Versatzkorrektur, während die Parameter zur Generierung der binären Maskierung unter *maskGenerator* zusammengefasst sind. Weitere Parameter zur eigentlichen Objekterkennung sind direkt unter der *septoSurvive* Ebene eingegliedert.

```
{
  "septoSurvive": {
    "imgHandler": {
      "imgWidth": 1920,
      "imgHeight": 1200,
      "dataPath": "../images/",
      "dataSet": "",
      "tMin": 0,
      "tMax": 15,
      "tStep": 1,
      "zMin": 0,
      "zMax": 120,
      "zStep": 1,
      "scale": 10.0
    },
    "resultsHandler": {
      "resPath": "results/"
    },
    "imageRegistrator": {
      "ambRatio": 0.95,
      "distL2": 20.0,
      "relShiftDev": 0.5
    },
    "maskGenerator": {
      "thr": 1.4,
      "morphSize": 15
    },
    "thrBZ": 0.7,
    "thrBTZ": 0.4,
    "thrB0": 0.6,
    "thrB1": 0.3
  }
}
```

Abbildung 15: Beispielhafte *.json* Konfiguration

1.1.3.1.5 Versatzkorrektur, Maskierung & Objekterkennung

Die nachfolgenden Unterkapitel beschreiben den eigentlichen algorithmischen Ablauf zur bildbasierten Wachstumsanalyse.

1.1.3.1.5.1 Versatzkorrektur

Wie in Kapitel 1.1.3.1.1 bereits beschrieben, beinhalten die zur Verfügung stehenden 4D-Datensätze aufnahmebedingte Verschiebungen in der xy-Bildebene. Um die spätere Aussagekraft der berechneten Wachstumskurven zu erhöhen, wird der beobachtbare Analysebildbereich so eingeschränkt, dass er in allen Zeitschritten sichtbar ist. Dazu wurde folgendes Vorgehen festgelegt:

1. Es wird angenommen, dass die affine Transformation zwischen zwei zeitlich benachbarten Bildern nur aus einer reinen Verschiebung besteht.
2. Zur Berechnung der Verschiebung zwischen zwei benachbarten Bildern werden Featurepunkte verwendet.
3. Abschließend werden alle ermittelten Bildverschiebungen zwischen zwei Zeitpunkten t und $t+1$ auf den gemeinsamen Bezugspunkt $t=0$ umgerechnet.
4. Die maximale Verschiebung in Bezug auf $t=0$ begrenzt den beobachtbaren Analysebildbereich.

Als Featurepunkte werden im Allgemeinen lokal auffällige Bildpunkte bezeichnet, die sich aufgrund ihres Wiedererkennungswertes u.a. zur Berechnung einer globalen Bildverschiebung eignen. Solch auffällige Bildpunkte können beispielsweise Eckpunkte oder Bildkanten sein. Für dieses Projekt wurden SIFT (scale invariant feature transform) Featurepunkte verwendet [2].

Für die Schätzung der Bildverschiebung zwischen zwei Zeitpunkten t und $t+1$ werden oben genannte Featurepunkte jeweils in beiden Bildern unabhängig voneinander detektiert. Im nächsten Schritt werden die beiden Featurepunktmenge aufeinander abgebildet, indem versucht wird Featurepunktpaare zu finden, die jeweils physikalisch identische Objektpunkte in den beiden Bildern beschreiben. Im Idealfall würden alle gefundenen Featurepunktpaare dieselbe örtliche Verschiebung aufweisen.

Zur Erhöhung der Matchingrobustheit wird der Parameter *ambRatio* verwendet, der die Eindeutigkeit eines Matches beschreibt. Je kleiner der Wert von *ambRatio* konfiguriert wird, desto eindeutiger muss die Zuordnung sein um verwendet zu werden. Die Qualität einer Zuordnung wird dabei über eine Vergleichsmetrik beschrieben, die die Ähnlichkeit zweier Featurepunkte bewerten kann. Eine Zuordnung ist dann eindeutig, wenn die Ähnlichkeit eines Featurepunktes zum besten Match deutlich höher ist als zum zweitbesten Match. Unter der Annahme, dass die Bildverschiebung begrenzt ist, sind außerdem nur Matches zugelassen, deren Featurepunkte nah genug aneinander liegen und somit die euklidische Distanz zwischen den Koordinaten einen Grenzwert (*distL2*) nicht überschreitet. Alle Featurepunktpaare, die oben genannte Bedingungen bezüglich *ambRatio* und *distL2* erfüllen, werden anschließend verwendet um eine mittlere Verschiebung zwischen den beiden Bildern abzuschätzen. Zur Erhöhung der Robustheit werden abschließend noch Featurepunktpaare aussortiert, deren Verschiebung eine zu große Abweichung zur errechneten mittleren Verschiebung aufweist. Als Parameter steht hierfür *relShiftDev* zur Verfügung, der die maximal erlaubte relative Abweichung in Bezug auf die maximale Abweichung eines aktuellen Featurepunktpaares zum berechneten Mittelwert vorgibt.

Die nachfolgende Abbildung 16 zeigt die detektierten Featurepunkte für zwei aufeinanderfolgende Zeitpunkte t und $t+1$ einer beliebigen Tiefenschicht in der oberen Bilderreihe. In der unteren Bilderreihe sind die gematchten Featurepunkte für den Zeitpunkt t abgebildet.

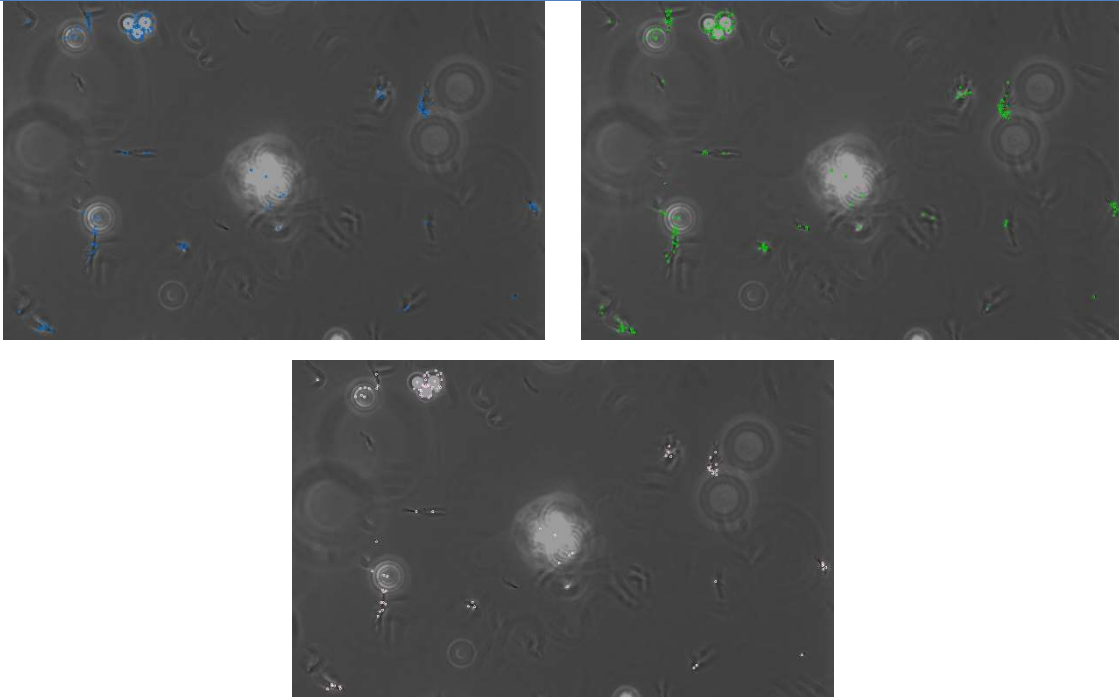


Abbildung 16: Featurepunkte für zwei aufeinanderfolgende Zeitpunkte t (oben, links) und $t+1$ (oben, rechts) und gematchte Featurepunkte für Zeitpunkt t (unten)

Die mittlere Verschiebung aller am Ende ausgewählten Featurepunktpaare beschreibt die gesuchte globale Verschiebung zwischen den beiden betrachteten Zeitschritten t und $t+1$.

1.1.3.1.5.2 Maskierung

Wie in Kapitel 1.1.3.1.2 bereits beschrieben, sind im beobachteten Nährmedium *Beads* eingelassen, um parallel die Sauerstoffkonzentration zu erfassen. Diese Strukturen sind damit auch in den aufgenommenen Bilddaten sichtbar und müssen entsprechend maskiert werden. Die automatisierte Erkennung dieser Fremdobjekte wird in diesem Kapitel betrachtet.

Im vorherigen Kapitel wurde definiert, dass die zeitliche Verschiebung zwischen den einzelnen Bildern in Bezug auf Zeitpunkt $t=0$ interpretiert wird. Folglich wird die binäre Maske zum Ausschluss der *Beads* ebenfalls für den Zeitpunkt $t=0$ definiert. Dafür wurde folgendes Verfahren festgelegt:

1. Für $t=0$, iteriere durch alle z -Schichten und bilde ein Minimumbild. Dabei enthält das Minimumbild am Ende pixelweise den kleinsten, über alle Tiefenschichten vorkommenden, Helligkeitswert.
2. Generiere eine Binärmaske durch pixelweise Bewertung der Abweichung der Helligkeit des Minimumbildes zum Mittelwert des Minimumbildes.

Die Abbildung 17 zeigt für einen *e_coli* Datensatz das errechnete Minimumbild für den Zeitpunkt $t=0$ (oben links), die generierte Binärmaske (oben rechts) sowie das resultierende maskierte Bild (unten).

Der Analysebildbereich ist bereits so begrenzt, dass er über den gesamten Zeitbereich sichtbar ist. Auffällig ist, dass im Minimumbild die einzelnen Objekte deutlich besser sichtbar sind, als in den unbehandelten Einzelbildern. Für das Minimumbild wird der mittlere Helligkeitswert berechnet (ohne Berücksichtigung der schwarzen Ränder). Anschließend wird jedes Pixel markiert, dessen Helligkeitswert um ein konfigurierbar Vielfaches höher ist als der errechnete Mittelwert. Dazu wird in der

Kategorie *maskGenerator* der Konfigurationsparameter *thr* verwendet. Zur Rauschunterdrückung und Auffüllen kleiner Löcher in der Binärmaske wird abschließend noch die morphologische Operation *closing* verwendet [3]. Um die Größe des Strukturelements einzustellen wird der Konfigurationsparameter *morphSize* gesetzt.

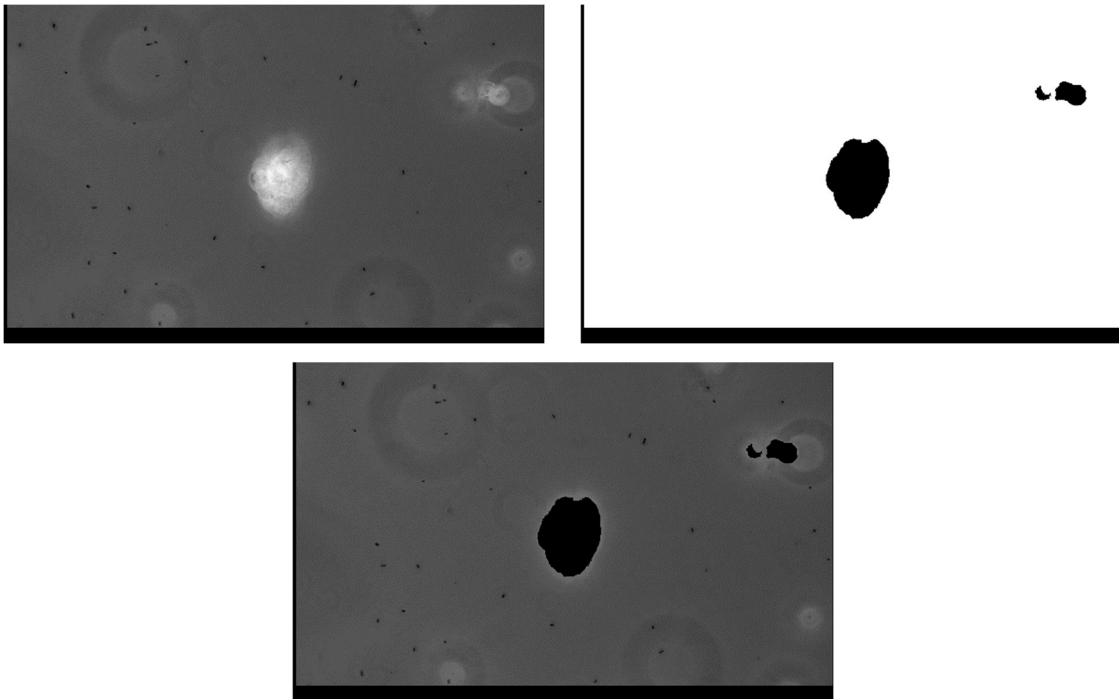


Abbildung 17: Minimumbild (oben, links), binäre Maske (oben, rechts), maskiertes Minimumbild (unten)

Die errechnete Binärmaske maskiert am Ende sowohl Bildbereiche, die aufgrund der globalen Bildverschiebungen nicht über den gesamten Zeitraum sichtbar sind, als auch die aufgrund ihrer auffälligen Helligkeitsstruktur erkannten Störobjekte (*Beads*).

1.1.3.1.5.3 Objekterkennung

Ziel der Objekterkennung ist es, die Bakterienanzahl pro Zeitschritt zu ermitteln, um das Wachstum entsprechend beschreiben zu können. Dazu wurde folgendes Vorgehen definiert:

1. Pro Zeitschritt: Generiere ein Minimumbild über alle vorhandenen Tiefenschichten
2. Gewichte anteilig Informationen aus der zeitlichen Vergangenheit
3. Detektiere Objekte durch pixelweise Bewertung der Abweichung der Bildhelligkeit vom Mittelwert
4. Nachbearbeitung mittels morphologischer Operationen (optional)

In einem ersten Schritt wird das Minimumbild für den aktuellen Zeitschritt berechnet. Damit enthält jedes Pixel seinen minimalen Helligkeitswert über alle vorhandenen z-Ebenen. Um robuster gegenüber zeitlichen Helligkeitsschwankungen zu sein, wird anschließend die bereits vorhandene Minimalinformation aus der Vergangenheit integriert. Dazu dienen die beiden Schwellwerte *thrBZ* und *thrBTZ*. Diese werden auf das absolute Differenzsignal zwischen aktuellem Minimumbild und

Minimumbild der Vergangenheit angewandt. Ist die Differenz größer als $thrBZ$, so wird die Information aus dem aktuellen Minimumbild für die weitere Analyse verwendet. Für Differenzen kleiner als $thrBTZ$ werden die Informationen aus dem Minimum aus aktuellem Minimumbild und Minimumbild aus der Vergangenheit verwendet. Für Werte zwischen $thrBZ$ und $thrBTZ$ werden die Signale entsprechend linear interpoliert.

Anschließend wird das aktualisierte Minimumbild für die Objekterkennung verwendet. Dazu werden erneut zwei Konfigurationsparameter $thrB0$ und $thrB1$ definiert. Ist ein Pixel dunkler als $thrB1$ -anteilig vom Mittelwert des aktualisierten Minimumbildes, so wird es als 100% Objektpixel erkannt. Ist ein Pixel heller als $thrB0$ -anteilig vom Mittelwert des aktualisierten Minimumbildes so wird es als 0% Objektpixel erkannt. Dazwischen wird wieder entsprechend linear interpoliert. Als optionaler Schritt kann abschließend wieder eine morphologische *closing* Operation angewendet werden um Löcher und Rauschen im Segmentierungsergebnis zu füllen beziehungsweise zu verringern.

Die nachfolgende Abbildung 18 zeigt beispielhaft ein berechnetes Minimumbild zum Zeitpunkt $t=20$ (links) und das entsprechende Segmentierungsergebnis der Objekterkennung (rechts).

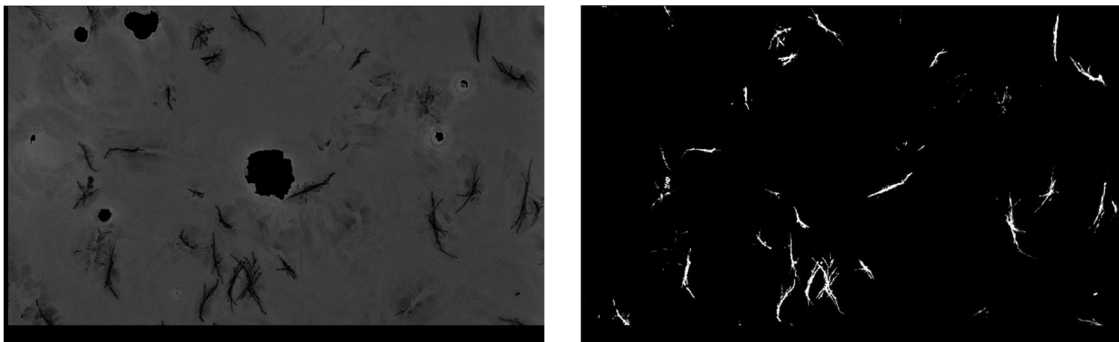


Abbildung 18: Minimumbild (links) und Segmentierungsergebnis (rechts)

Um eine Aussage über die gesuchte Bakterienanzahl zu einem bestimmten Zeitpunkt treffen zu können werden einfach die Pixelwerte im Segmentierungsergebnis aufsummiert.

1.1.3.1.6 Results Handler

Die Aufgabe des Results Handler Moduls ist es, die Objekterkennungsergebnisse so aufzubereiten, dass eine Bewertung und Analyse des Wachstumsprozesses möglich ist.

Ein charakteristischer Punkt zur späteren Bewertung ist der Umkehrpunkt des Wachstumsprozesses. Um diesen zu ermitteln werden die Ergebnisse mit einer Sigmoid-Funktion nach

$$y = \frac{L}{1 + e^{-k*(x-x_0)}} + b$$

angenähert. Dabei beschreibt L den Maximalwert, k die Steigung, und x_0 und b zwei Verschiebungswerte auf der x beziehungsweise y Achse. Der Wendepunkt des Wachstumsprozesses korrespondiert dabei mit dem Verschiebungswert x_0 .

Die nachfolgende Abbildung 19 zeigt die Objekterkennungsergebnisse für das *E_coli* Bakterium zu den beispielhaften Zeitpunkten $t=5$, $t=10$, $t=15$ und $t=20$. Das Bakterienwachstum ist in den Segmentierungsergebnissen deutlich sichtbar. Wie bereits beschrieben wird der Wachstumsprozess für eine kompakte Analyse mit einer sigmoidalen Funktion approximiert.

Die resultierende kompakte Kurvenbeschreibung ist in Abbildung 20 dargestellt. Auf der linken Seite ist die normierte Wachstumskurve abgebildet. Aus dieser ist der Wachstumsfaktor (für dieses Beispiel ca. 27.0) ablesbar. In der mittleren Grafik ist die sigmoidale Approximation der Wachstumskurve zu sehen.

Aus dieser Beschreibung lässt sich der Wendepunkt als charakteristische Eigenschaft des Wachstumsprozesses ermitteln. Für dieses Beispiel liegt der zeitliche Wendepunkt bei $t=13.38$. Auf der rechten Seite ist die relative Wachstumskurve als erste zeitliche Ableitung aufgezeigt. Auch hier ist der Umkehrpunkt deutlich sichtbar. Im Sättigungsbereich der Kurve geht das relative Wachstum entsprechend gegen Null.

Abbildung 21 zeigt abschließend die entsprechenden Wachstumskurven für das *E_coli* Bakterium unter Zugabe von Antibiotikum. Es ist deutlich sichtbar, dass das Wachstum stark eingeschränkt ist. In diesem Fall liegt der Wachstumsfaktor nur noch bei ungefähr 1.75.

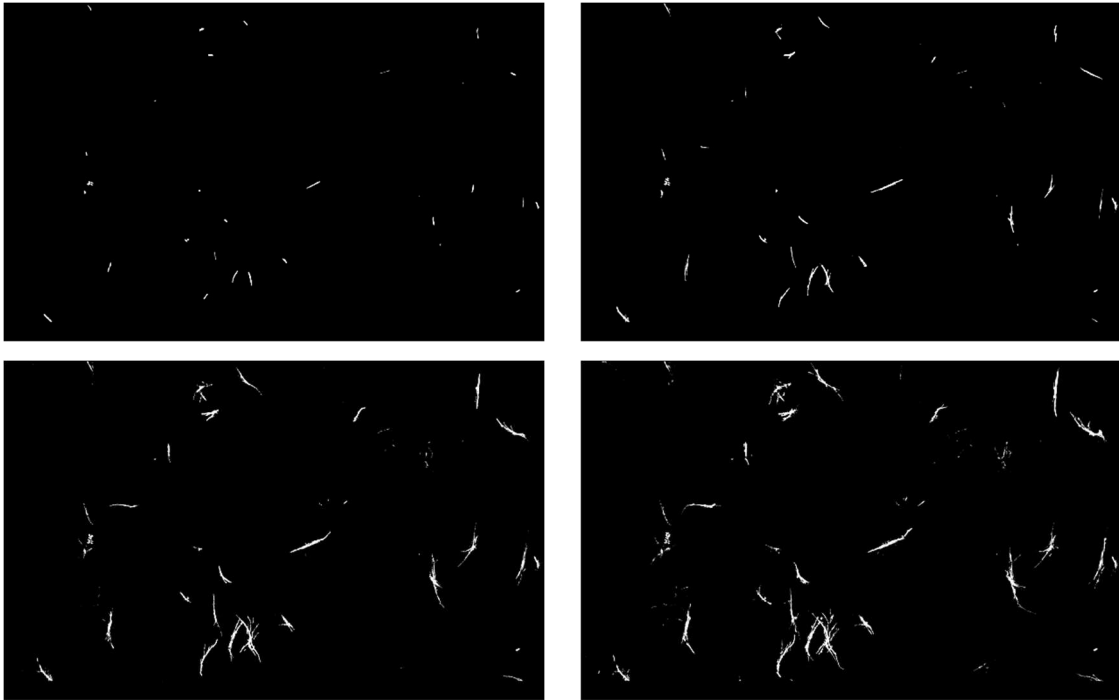


Abbildung 19: Wachstumserkennung für *e_coli* zu den Zeitpunkten $t=5$ (oben, links), $t=10$ (oben, rechts), $t=15$ (unten, links) und $t=20$ (unten, rechts)

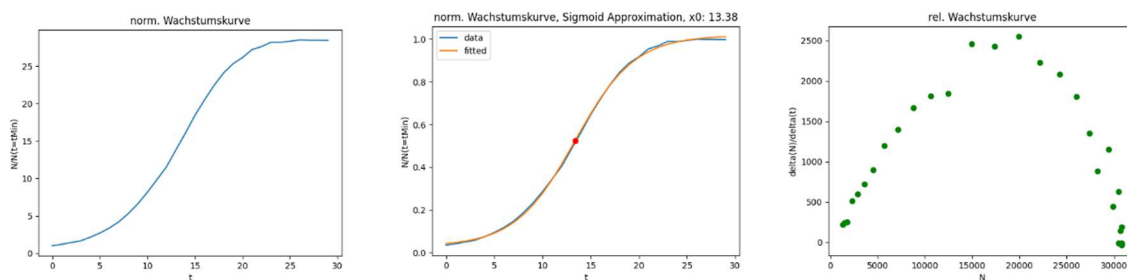


Abbildung 20: *E_coli*: normierte Wachstumskurve (links), sigmoidale Annäherung (Mitte) und relative Wachstumskurve (rechts)

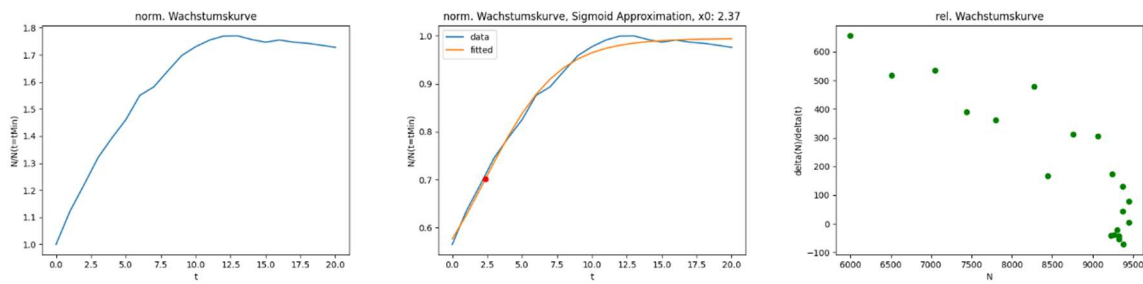


Abbildung 21: *E. coli* unter Zugabe von Antibiotikum: normierte Wachstumskurve (links), sigmoidale Annäherung (Mitte) und relative Wachstumskurve (rechts)

Die errechneten Wachstumskurven können zur Plausibilisierung noch mit Sauerstoffkurven verglichen werden, die mithilfe eines Messsystems von *ibidi* aufgenommen werden. Die nachfolgende Grafik zeigt beispielhaft den gemessenen Sauerstoffverbrauch für das *E. coli* Bakterium für drei verschiedene Positionen auf der linken Seite und die ermittelte Wachstumskurve für Position 3 auf der rechten Seite. Die Sauerstoffkurve ist dabei auf die ersten 21 Messpunkte beschränkt. Nach 4 Stunden (entspricht Messpunkt $t=14$ in der rechten Abbildung) ist die Sauerstoffkonzentration auf 0% abgefallen. Die Wachstumskurve befindet sich zu diesem Zeitpunkt bereits hinter dem Umkehrpunkt. Anschließend tritt die Sättigungsphase ein.

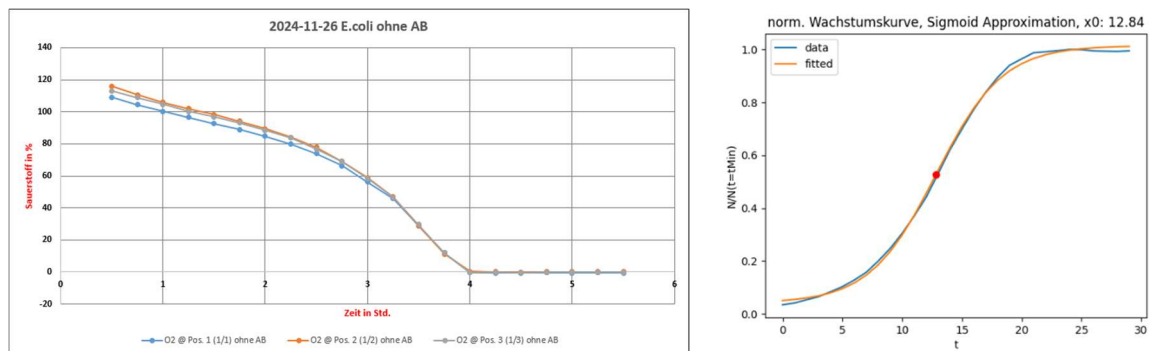


Abbildung 22: Vergleich: OPAL Sauerstoffmesskurve (links) und ermittelte Wachstumskurve (rechts)

1.1.4 AP SOL5: Aufbau eines Demonstrators zur Konzeptentwicklung eines Gesamtsystems

In gemeinsamer Abstimmung hat das Konsortium mit den Vertretern des Projektträgers entschieden die Herstellung eines funktionsfähigen Gesamtdemonstrators, in dem alle Teilprojekte integriert werden, zu verwerfen.

Als Ersatz dient die unabhängige Demonstration der Funktionalität der Teilkomponenten. Zwischen den Teilkomponenten sind manuelle Transferschritte nötig

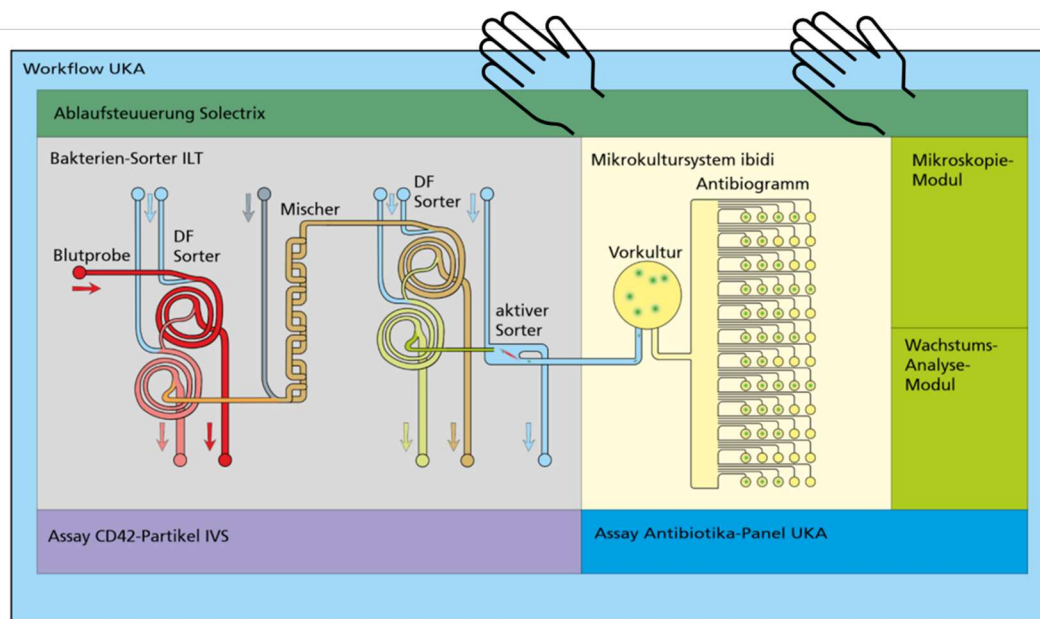


Abbildung 23: Demonstratorkonzept mit Teillösungen

1.1.5 AP SOL6: Konzept zur Erkennung von antibiotischen Resistenzprofilen aufgrund von Wachstumskinetiken

Die minimale Hemmkonzentration (MHK) gibt die minimal benötigte Konzentration eines Antibiotikums an, die benötigt wird, um das Bakterienwachstum in einer kontrollierten Laborumgebung zu stoppen. Sie ist damit ein wichtiger Indikator, um die Wirksamkeit eines Antibiotikums zu beschreiben.

Die in obigen Arbeitspaketen beschriebene algorithmische Schätzung der bakteriellen Wachstumskurven eignet sich dabei wie folgt für die Erkennung algorithmischer Resistenzprofile.

Im Folgenden werden die ermittelten Wachstumskurven für das *e_coli* Bakterium bei unterschiedlichen Antibiotikakonzentrationen gegenübergestellt. Die folgende Abbildung 24 zeigt in der ersten Bilderreihe drei initiale Bakterienlösungen bei $t=0$ für das *e_coli* Bakterium ohne Zugabe von Antibiotikum (links) mit Zugabe von Antibiotikum (rechts) und mit einer 1:10 verdünnten Antibiotikakonzentration (Mitte). Die zweite Bilderreihe zeigt das jeweilige Detektionsergebnis zum Zeitpunkt $t=30$. Es ist klar erkennbar, dass die Zugabe von Antibiotikum eine stark wachstumshemmende Wirkung hat, die bei der verdünnten Lösung etwas schwächer ausfällt. Das spiegelt sich auch in den entsprechenden Wachstumskurven wider, die in der dritten Bilderreihe gezeigt sind. Während der Wachstumsfaktor ohne Antibiotikazugabe bei etwa 30.0 liegt, fällt er für die verdünnte Antibiotikazugabe auf etwa 10.0 ab. Im unverdünnten Fall liegt der geschätzte Wachstumsfaktor nur noch bei etwa 3.0. Auffällig ist auch, dass die Glattheit der Wachstumskurven bei Antibiotikazugabe geringer wird. Dies liegt an der teilweisen Deformation der einzelnen Bakterien durch das Antibiotikum und eine Kontrastschwankung, die sich direkt in den Detektionsergebnissen widerspiegelt.

Über den ermittelten Wachstumsfaktor lässt sich die Wirksamkeit eines Antibiotikums gut bewerten. Die MHK lässt sich dabei durch ein schrittweises Absenken und gleichzeitiges Beobachten des Wachstumsfaktors ermitteln. Definiert man einen oberen Grenzwert für den gerade noch erlaubten Wachstumsfaktor, so ergibt sich die MHK als geringste Antibiotikamenge, bei der der Grenzwert nicht überschritten wird.

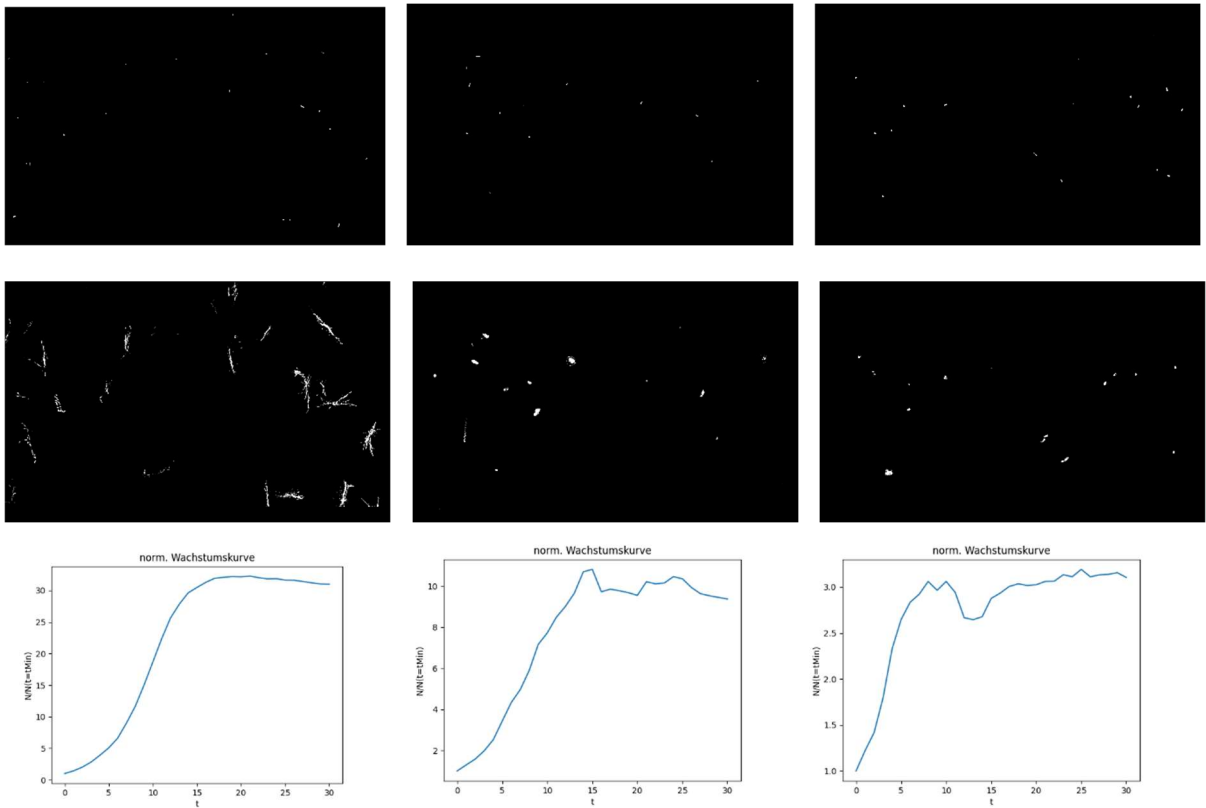


Abbildung 24: Detektionsergebnisse und Wachstumskurven ohne Antibiotikazugabe (links), mit Antibiotikazugabe (rechts) und verdünnter Antibiotikazugabe (Mitte)

2 Literaturverzeichnis

- [1] „<https://pythonhosted.org/python-bioformats/>,“ [Online].
- [2] D. G. Lowe, „Object Recognition from Local Scale-Invariant Features,“ *Proceedings of the International Conference on Computer Vision (ICCV)*, Bd. 2, pp. 1150-1157, 1999.
- [3] „https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html,“ [Online].