alaagi™

PROJECT BRIEF

Seaweed Flaw & Order

Build the software for a scanner that inspects bioplastic film.

THE PROBLEM

Every sheet of film we make has to be judged by eye. Is it cloudy? Are there bubbles? Is it the same thickness across the sheet? Right now a person looks at it and decides. That is slow, it is inconsistent between people, and it tells us nothing about why a sheet failed. Factories buy $50,000 camera systems for this, but those inspect film moving through a production line at speed. We make one sheet a day in a lab.

WHAT YOU BUILD

A desktop application that drives a simple scanner rig and grades a sheet of film in under a minute. The rig is an LED light pad, a camera, a few sensors on a moving head, and an Arduino. You write the software: the interface, the image analysis, the device control, and the record keeping.

IT SHOULD MEASURE

- **Defects**
  Bubbles, specks, cracks, found, counted and mapped
- **Clarity and haze**
  How much light passes straight through vs scatters
- **Thickness uniformity**
  Where the sheet is thicker or thinner
- **Colour**
  Yellowing or discolouration between batches

RULES

Any sheet size

From A4 down to a small test coupon

Runs offline

Everything stays local. No cloud, no uploads.

Honest output

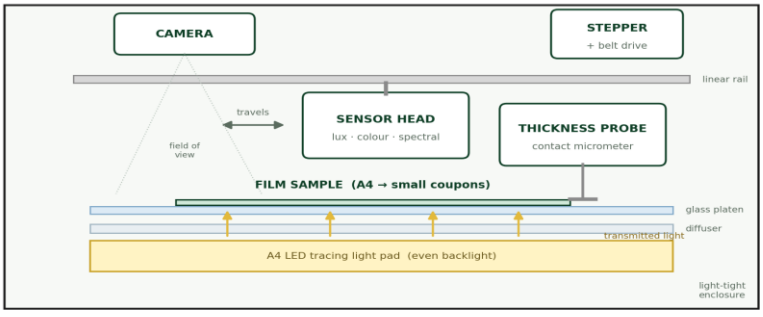
Say how confident it is, and show the evidence

Usable by anyone

A lab tech should run it without a manual

THE RIG YOU ARE CODING FOR

Figure 1 — Scanner rig, side view



DELIVERABLE

A working application we can install and use, the source code, and a short note on how to calibrate it and what its limits are.

Questions or ideas: **sheheryar.khan@smu.ca** Hardware list available on request

alaagi™

Seaweed Flaw & Order- Hardware Guide

A scanner rig for inspecting seaweed bioplastic film · companion to the project brief

This document describes the hardware the software will control. You are not required to build the rig - we will assemble it - but the software needs to talk to it, so it helps to understand what each part does and what data it produces.

The short version: think of a flatbed scanner. A sheet of film lies on glass above an even light source. A camera photographs it from above, and a small sensor head travels across on a rail taking point measurements. An Arduino reads the sensors and drives the motor; your software does everything else.

Figure 1 — Scanner rig, side view

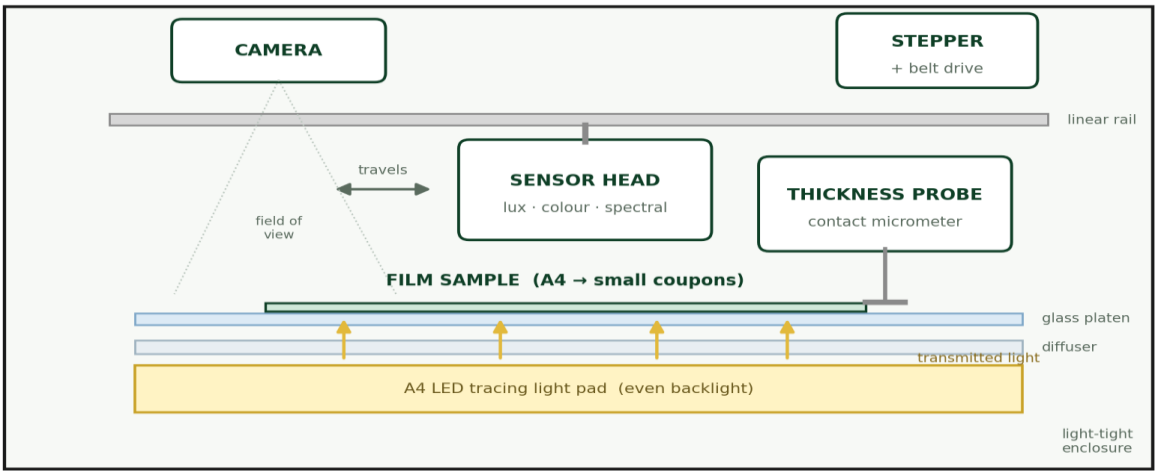


Figure 1 - The rig in side view. Light passes up through the film; the camera and sensor head read what comes out the other side.

How each measurement works

Everything here comes down to one idea: shine known light through the film and measure what emerges. A perfect film passes light cleanly and evenly. Every defect changes that in a measurable way.

Figure 2 — What each measurement actually measures

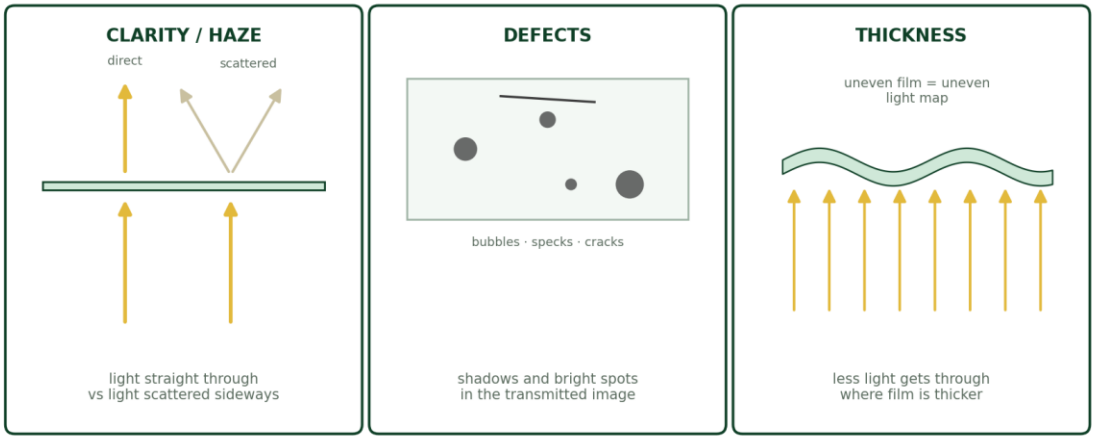


Figure 2 - The three core measurements.

Defects

With an even backlight, a bubble, speck or crack blocks or bends light and appears as a dark or bright spot in the photograph. Finding them is a blob detection problem: threshold the image, find connected regions that differ from their surroundings, then measure each one's size and position. The useful output is not just a count but a map, so we can see whether flaws cluster at the edges, the centre, or along the casting direction.

Clarity and haze

Clarity is how much light passes straight through. Haze is how much is scattered sideways on the way. A clear film and a cloudy film can transmit the same total amount of light, so total brightness alone is not enough; the ratio between direct and scattered light is what matters. This is why there is a light sensor directly under the film and the camera above it: comparing the two gives a usable haze estimate.

Thickness uniformity

Where film is thicker, less light gets through. Once calibrated against samples of known thickness, the brightness map from the camera becomes a thickness map. This is a relative measurement, not an absolute one, which is why the rig also carries a contact probe to give true readings at a few points and anchor the calibration.

Colour

Our film can yellow or shift tone between batches. A colour sensor reading transmitted light gives numeric colour values, so two batches can be compared objectively rather than by eye.

The parts list

All of the items below are findable on Amazon.ca. Prices are rough and for guidance only.

Core

Part	What it does	Approx.	Search term
Arduino Uno R4 WiFi	Reads the sensors, drives the motor, talks to your software over USB	$40	arduino uno r4 wifi
Elegoo UNO starter kit	Cheaper alternative, includes breadboard, wires and components	$60	elegoo uno r3 starter kit
Laptop or Raspberry Pi 5	Runs your application and handles the camera. Image processing is too heavy for an Arduino.	- / $120	raspberry pi 5 4gb
USB webcam or Pi Camera 3	Photographs the sheet. Autofocus and 1080p or better.	$60-100	logitech c920 / raspberry pi camera module 3

Lighting

Part	What it does	Approx.	Search term
A4 LED tracing light pad	The even backlight the whole method depends on. Dimmable is better.	$30	a4 led tracing light pad
Diffuser sheet	Evens out any remaining hot spots. White acrylic or drafting vellum.	$15	white translucent acrylic sheet a4
Black foam board	Blocks room light. The enclosure matters more than people expect.	$15	matte black foam board
Linear polarising film (x2)	Optional. Crossed polarisers reveal internal stress patterns invisible in normal light.	$20	linear polarizing film sheet

Sensors - all I²C, all talk to the Arduino on the same two wires

Part	What it does	Approx.	Search term
TSL2591 light sensor	Wide-range lux readings for transmission and haze	$15	adafruit tsl2591
TCS34725 colour sensor	RGB colour of transmitted light, yellowing and tone	$18	tcs34725 rgb color sensor
AS7341 spectral sensor	Eleven wavelength channels instead of three. The best single upgrade if budget allows.	$30	as7341 spectral sensor
SHT31 temp / humidity	Logs room conditions, which affect drying and film behaviour	$15	sht31 temperature humidity sensor
Qwiic / STEMMA QT cables	Plug sensors together without soldering	$15	stemma qt qwiic cable kit

Motion

Part	What it does	Approx.	Search term
NEMA 17 stepper motor	Moves the sensor head across the sheet	$20	nema 17 stepper motor
A4988 driver + CNC shield	Lets the Arduino drive the stepper	$20	cnc shield v3 a4988
Linear rail kit, 400 mm	The track the head rides on	$50	sbr12 400mm linear rail
GT2 belt and pulleys	Connects motor to head	$15	gt2 timing belt pulley kit
Endstop switches	Tells the software where “home” is	$10	mechanical endstop limit switch

Thickness and calibration

Part	What it does	Approx.	Search term
Digital micrometer	True thickness at a point. Models with a data port can feed the Arduino directly.	$40–200	digital micrometer data output
Feeler gauge set	Known thicknesses for calibrating the light-based estimate	$15	feeler gauge set metric
Plastic shim assortment	Film-like samples of known thickness	$20	plastic shim stock assortment

Figure 3 — How the parts connect

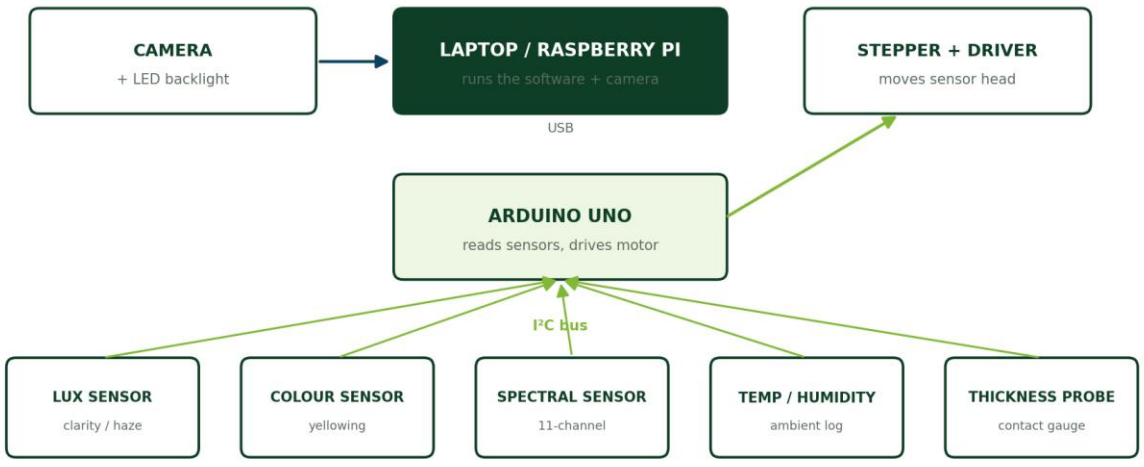


Figure 3 - How the parts connect. The Arduino handles sensors and motion; your computer handles the camera, the analysis and the interface.

What the software needs to do

This is the actual deliverable. The hardware is a means to an end.

Figure 4 — Suggested software interface

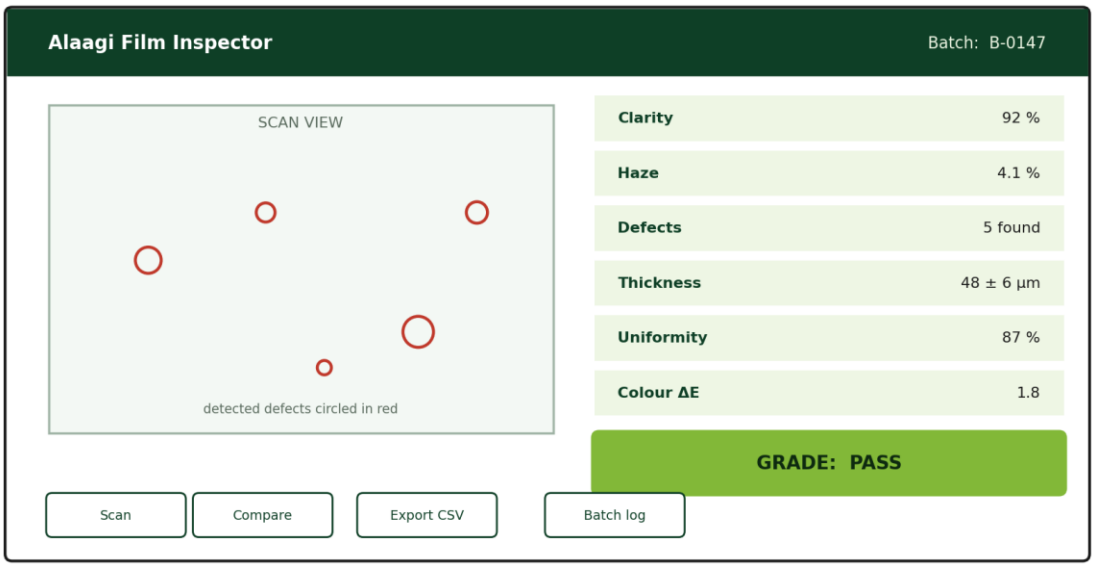


Figure 4 - A suggested layout. Yours does not have to look like this; it has to be usable by someone in a lab coat in a hurry.

Core functions

- Connect to the rig, show device status, and fail gracefully when something is unplugged
- Run a scan: enter a batch ID, capture, analyse, grade, save, ideally in under a minute
- Report defects: count, size, position, and a visual map overlaid on the image
- Report clarity, haze, colour values and a thickness uniformity map
- Grade pass or fail against thresholds that we can set and change ourselves
- Calibrate: a guided routine using a blank scan and known-thickness samples
- Keep a batch log, let us compare two scans side by side, and export to CSV

Constraints that matter

- Any sheet size from A4 down to a coupon a few centimetres across. Detect the sample edges or let the user drag a box.
- Runs entirely offline on a local machine. Nothing uploads anywhere.
- Shows uncertainty. A thickness estimate from brightness is an estimate, say so, and say what it was calibrated against.
- Repeatable. Scanning the same sheet twice should give the same answer. Build a test for this.

Suggested stack

Python with OpenCV for image analysis, pyserial for the Arduino link, SQLite for the batch log, and whatever you like for the interface, a desktop toolkit or a local browser app both work. Nothing here requires a cloud service, and nothing should use one.

What we provide

- The rig, assembled, with all sensors wired and a basic Arduino sketch that reports readings over serial
- Real film samples, including deliberately flawed ones, so you have good and bad examples to work with
- Reference measurements from our existing equipment so you can check your results against something real
- Access to us for questions throughout

Contact:

Sheheryar Khan

Founder & CEO

Alaagi Inc.

sheheryar.khan@smu.ca