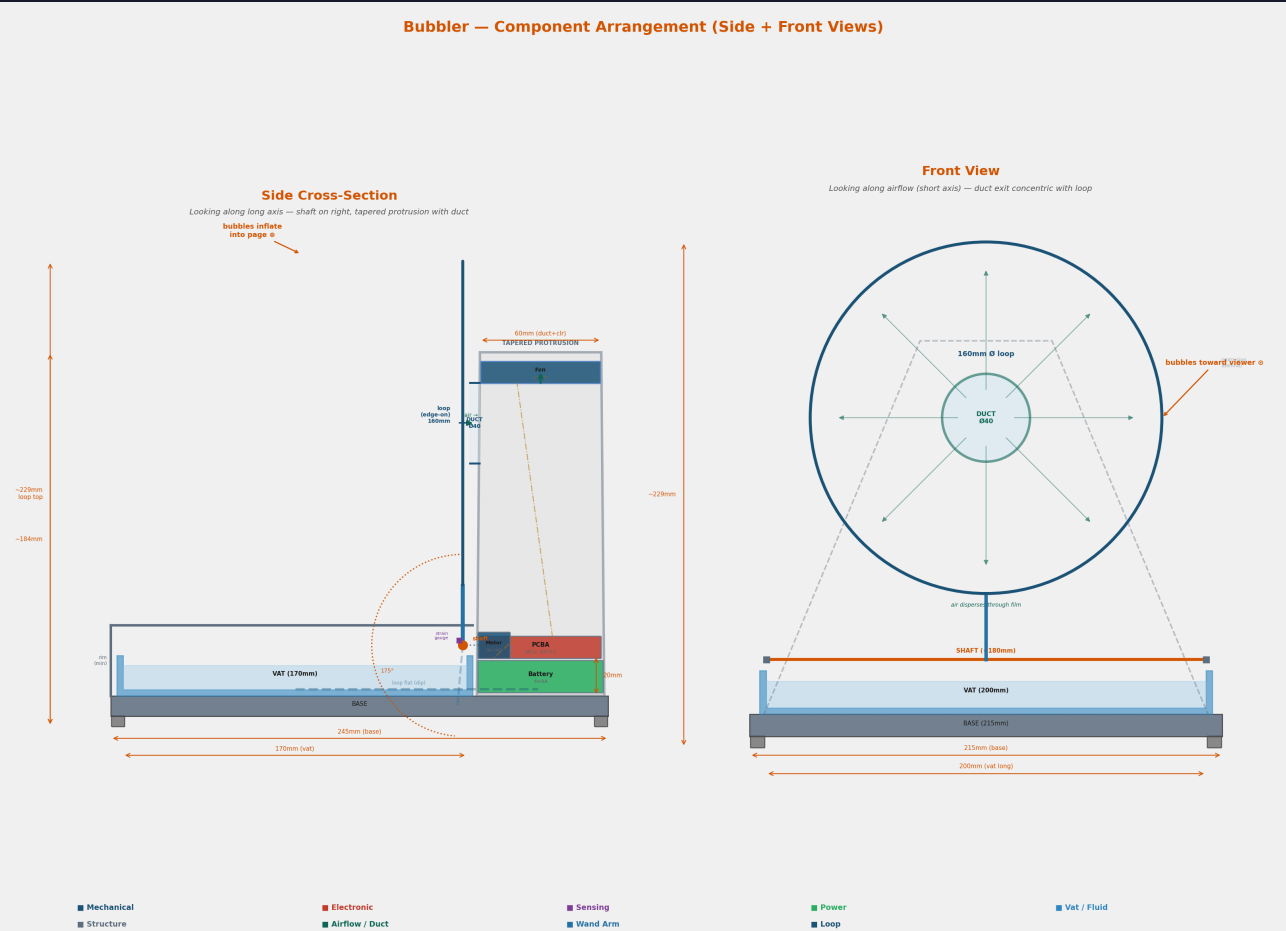


Bubbler

Big bubbles. Zero effort.

An automated soap bubble machine that produces giant bubbles up to 500 mm using force-curve feedback. No app, no cloud -- just press power and watch.



The Problem

Large bubbles are hard to produce

Giant soap bubbles (200-500 mm) require precise airflow control, correct dip timing, and film that survives rotation. Manual technique is inconsistent and takes practice to learn.

Existing machines make only small bubbles

Consumer bubble machines produce 20-50 mm bubbles with high pop rates and no adaptation. They blow hard, pop fast, and waste soap. No feedback loop means no improvement over time.

Gap between cheap toys and pro gear

Toy machines cost \$10-30 but produce tiny bubbles. Professional stage equipment costs \$200+ and needs power outlets. Nothing in between serves families, performers, and outdoor events well.

TARGET USERS

Families | Performers | Event organizers

Outdoor parties | Buskers | Kids' entertainment

How It Works

1 Fill

Fill vat with soap solution

Pour bubble solution into the built-in vat. Standard dish soap mix works. The wand loop sits submerged, ready to dip.

2 Press

Press power -- wand dips into vat

Single button press starts the cycle. The motor arm dips the wand loop into the soap solution, coating it with a thin film.

3 Inflate

Arm rotates up, fan gently inflates film

The arm rotates 175 degrees upward. A DC fan blows a controlled ramp of air through the soap film, inflating it into a large bubble.

4 Optimize

Force sensing optimizes each cycle

A strain gauge on the wand arm measures 50-200 mN during inflation. Firmware classifies each outcome and auto-adjusts fan speed, dip duration, and blow ramp. Converges in 5-10 cycles.

Architecture

Sensing Signal Chain

Strain gauge --> HX711 ADC --> STM32 MCU

50-200 mN force 24-bit 10-80 Hz hill-climbing optimizer

Actuation Chain

STM32 --> Motor H-bridge (dip/rotate) + Fan PWM (inflate)

KEY COMPONENTS

MCU (STM32)

Bare-metal firmware, hill-climbing on 5 parameters, no app/cloud

Strain Gauge + HX711

Force-curve feedback at 10-80 Hz, cycle outcome classification

DC Motor + H-bridge

Wand dip and 175-degree arm rotation, bidirectional control

DC Fan + MOSFET

PWM-controlled airflow ramp for gentle bubble inflation

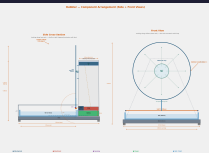
Power (4xAA batteries)

~7 hr runtime, simple replacement, no charging needed

Enclosure (IPX4)

215x206 mm footprint, ~250 mm height, splash-resistant

Component Arrangement



Key Innovation

Force-curve feedback: the strain gauge on the wand arm turns every inflation cycle into a learning opportunity.

What It Measures

Force: 50-200 mN during inflation

Sample rate: 10-80 Hz via HX711 ADC

Resolution: sub-mN with 24-bit ADC

Cycle Outcome Classification

- Success:** Bubble detaches cleanly -- force drops to baseline
- Pop:** Sudden force spike then zero -- film burst mid-inflation
- No film:** Near-zero force throughout -- dip failed to coat
- Partial:** Force plateau then slow decay -- bubble formed but small

Hill-Climbing Optimizer

- **Fan speed ramp:**
PWM duty cycle profile during inflation
- **Dip duration:**
How long the wand stays in the soap vat
- **Blow ramp rate:**
How quickly airflow increases
- **Rotation speed:**
Arm angular velocity during lift
- **Pause duration:**
Wait time between dip and blow

Constraints & BOM

Bubble size	Up to 500 mm
Wand loop geometry + controlled airflow	
Battery life	~7 hrs (4xAA)
Low-power MCU, motor duty-cycled	
Footprint	215 x 206 mm
Compact enough for a tabletop	
Operating temp	5-40 C
Soap film physics limit the range	
Wind tolerance	4 kph crosswind
Fan ramp compensation algorithm	
Electronics	IPX4
Splash-resistant enclosure for outdoor use	

BOM ESTIMATE (~\$12.80 total)		
STM32 MCU		\$1.50
Strain gauge + HX711 ADC		\$0.80
DC motor + H-bridge		\$1.80
DC fan + MOSFET driver		\$1.10
Enclosure (molded plastic)		\$3.50
Wand + loop + shaft		\$1.50
Misc (PCB, connectors, passives)		\$2.60
Total BOM	~\$12.80 Target retail: sub-\$50	

4xAA batteries not included in BOM. No app, no cloud -- keeps ongoing costs at zero.

Hardest Problems

1 Film survival during rotation

The soap film must survive a 175-degree arm rotation from the vat to the blow position without breaking. Film thickness, rotation speed, and acceleration profile all matter. Too fast and the film tears from inertia; too slow and it drains and thins. Requires empirical tuning of the motor ramp curve.

2 Force-curve interpretation

Classifying cycle outcomes (success, pop, no-film, partial) from noisy strain gauge signals at 10-80 Hz. The HX711 output includes mechanical vibration, motor coupling, and wind noise. Signal conditioning and threshold-based classification must be reliable enough for the hill-climbing optimizer to converge.

3 Wind compensation at 4 kph

Outdoor use means crosswind. A 4 kph breeze changes the effective airflow through the soap film, shifting optimal fan speed and blow duration. The optimizer must detect wind-induced pop patterns and compensate within a few cycles. No wind sensor -- inferred from force-curve anomalies only.

All three require physical prototyping. Soap film behavior cannot be fully simulated -- build, measure, iterate.

Gate Result & Next

GATE: PASS

62 pass / 23 N/A / 3 minor

System description complete.

Ready to proceed to PRD.

KEY SPECS

Bubbles up to 500 mm | 4xAA (~7 hr) | 215x206 mm footprint
~250 mm height | IPX4 | 5-40 C | BOM ~\$12.80 | Sub-\$50 retail

ARCHITECTURE

Strain gauge + HX711 + STM32 | Bare-metal FW | No app, no cloud

3 MINOR GAPS

- Gap 1** FW versioning scheme not yet defined
- Gap 2** Decision consequences formatting incomplete
- Gap 3** Schedule milestones need dates

WHAT'S NEXT

- Build a functional prototype with off-the-shelf motor, fan, and HX711 breakout
- Validate soap film survival during arm rotation (175 deg)
- Test force-curve classification accuracy across soap formulations
- Measure wind compensation convergence at 4 kph crosswind
- Confirm 4xAA battery life target with real duty cycles

Next step: build the mechanical prototype. Validate film survival, force sensing, and the optimization loop before committing to PCB.