Spirometer Project Progress and Development Report

Prepared for project documentation and progress review

Prepared by: Akhmadjonov Abdulbosit

April 2026

Project focus	Gamified incentive spirometer for pediatric respiratory therapy
Main platform	Raspberry Pi 4, 3.5-inch LCD touch screen, flow sensor, Python game
Current position	Working game completed, first calibration formula kept in code, calibration rechecked against game behavior

Executive Summary

This report summarizes the main work completed for the spirometer project, starting from basic hardware setup and ending with a finished game that uses the calibrated sensor signal. The purpose of the project is to make respiratory therapy more engaging for children by combining a flow-based breathing device with a simple video game. The original design direction in the master project document was a self-contained Raspberry Pi spirometer with on-screen gameplay, data collection, and a child-friendly handheld body. The design was intended for children around ages 6 to 12 and was planned to collect therapy duration, FEV1/FVC-related values, and time when breathing dropped below the desired range.

On the hardware side, the Raspberry Pi, LCD touch screen, and flow sensor were successfully brought into operation. On the software side, the project moved through sensor readout, filtering, validation, game framework, scoring, therapy session logic, data logging, and settings with calibration. A major design change also happened in the game concept. The earlier Flappy Bird style direction was good for a fun idea, but it was not a good match for real therapy because therapy needs a longer and steadier breath. For that reason, the project changed to a car-based game where steady blowing keeps the player in the correct zone.

Calibration was done by comparing pulse frequency from the flow sensor with anemometer air-speed measurements in an 18 mm tube. That work produced the equation Flow (L/min) = max(0, 0.503 x Hz - 29.70), and this same first calibration formula was kept in the game code. Later, after the game was finished, calibration was revisited again to make sure the formula and the game behavior still agreed.

1. Project Background and Purpose

The project was created to answer a simple problem. Normal incentive spirometry can feel repetitive and boring, especially for children, so it is hard to keep them focused during the full therapy time. The main idea was to build a device that is not just a measurement tool, but also something interactive and easier for a child to use. The master design work described a tablet-like device with integrated screen, flow sensing, and internal electronics, instead of depending on a separate phone app.

From the physical design side, the selected direction was a curved handheld body with built-in grips, front hose placement, ventilation, and a protective plexiglass layer over the screen. During later rework, the housing was improved again by adding a dedicated slot for the flow sensor, more room for wires and the battery, a slot for plexiglass insertion, and deeper thumb areas to improve the hold. This gave a better match between the design on paper and the way the real parts had to fit together.

At the same time, the software goals were also clear. The system had to read the sensor reliably, convert the pulses into meaningful flow information, give the patient live feedback, run a therapy game, and save session information for review later.

2. Development Timeline Summary

Table 1 shows the main coding milestones that were used as project checkpoints. The percentage values are the progress marks from the internal timeline at that moment. After these checkpoints, more work was still done, and the game was later brought to a working finished state.

Code	Task	Checkpoint date	Timeline status
3.1	Get Raspberry Pi working	1/30/26	100%
3.2	Get LCD screen working	2/4/26	100%
3.3	Wire and test flow sensor	2/4/26	100%
3.4	Sensor readout driver	2/4/26	100%
3.5	Filtering and smoothing	2/10/26	100%
3.6	Sensor validation screen	2/16/26	100%
3.7	App framework	2/19/26	100%
3.8	Game with scoring	2/25/26	80%
3.9	Therapy session logic	3/10/26	60%
3.10	Data logging	3/19/26	50%
3.11	Settings and calibration	3/25/26	50%

3. Hardware Integration and Embedded Setup

The early stage of the work was mainly about making sure the basic electronics could communicate correctly. The Raspberry Pi was brought up first, then the LCD touch screen was made operational, and after that the flow sensor was wired and tested. This stage was important because without a stable sensor signal and a usable display, the rest of the project could not move forward in a meaningful way.

One of the practical achievements was getting the small 3.5-inch screen to work well enough for direct use on the device. This was not just a normal display task. The small size created real interface limits, so screen layout, game boundaries, and object visibility had to be adjusted. The project also included work on how to switch between the small LCD and HDMI when needed for testing and debugging.

Another important usability step was setting the Raspberry Pi to open the game menu automatically after boot. That made the device feel more like a dedicated product instead of just a development board running code from the terminal. This kind of small system behavior matters a lot in a medical or therapy-related device, because it reduces setup friction for the user.

The physical housing work from the master document also supported the code development. The selected concept used a one-body grip design, front hose placement, internal battery and Raspberry Pi cavity, ventilation openings, and a sliding back panel. Later rework improved part fit by adding a flow-sensor slot, more internal space, and better protection and ergonomics.

4. Sensor Readout, Filtering, and Calibration

After the hardware was active, the next large task was to turn the raw flow sensor signal into a useful software value. The flow sensor does not directly give airflow in liters per minute. It gives electrical pulses. Because of this, a readout driver had to be made first, then filtering and smoothing had to be added so the signal would not jump too much from noise or pulse timing changes.

A sensor validation screen was then created so the readings could be watched directly while testing. This was helpful because it gave immediate feedback and made it easier to see if the sensor was

reading at all, if it was too noisy, or if the smoothing logic was too aggressive. It also helped connect the software values to real blowing behavior before the game was fully done.

Calibration was done with an anemometer used as the reference device. Airflow was passed through an 18 mm tube, and for each test point the team recorded anemometer velocity, pulse frequency from the sensor, and applied pressure. The calibration document explains that the readings were collected across multiple flow levels, then matched as pairs of Hz and air velocity. From there a best-fit line was created to convert sensor frequency into velocity, and then the tube geometry was used to convert velocity into volumetric flow.

The final first calibration relation was Velocity (m/s) = 0.0329 x Hz - 1.95. After the tube conversion, this became Flow (L/min) = 0.503 x Hz - 29.70, with negative values clamped to zero in software. This equation was not only used for documentation. It was also kept in the game code itself, so the same calibration basis was used for gameplay behavior and therapy measurement.

Coming back to calibration after the game was finished was a good step. It helped verify that the game was still using the expected formula and that the interaction on screen was based on the same physical conversion that was developed during the first calibration work.

5. Game Development and Therapy Logic

The game development went through more than one stage. In the earlier project direction, the game idea followed the same general style as Flappy Bird. That idea made sense at first because it is simple and easy to understand, and it was also written in the master document as the original game concept. But later testing and discussion showed that it was not the best match for the therapy process.

The reason for the change was practical. A Flappy Bird type game naturally pushes the player toward many short breath actions, while respiratory therapy in this project needs a longer and steadier exhale. Because of this mismatch, the design pivoted into a car game. In the newer game, steady blowing keeps the car in the desired lane or zone, while weak or inconsistent breathing causes the car to drift away. This was a better connection between medical purpose and gameplay.

After the pivot, more therapy logic was added. The work included a game framework, scoring, therapy session control, and later additions related to holding the breath effort in the correct range. The project also explored how to count useful blowing time, how to give the user rest periods, and how to make the screen show feedback that was easy to understand. These changes made the project less like a normal arcade game and more like a guided therapy session with game motivation.

The final result of this phase was a working finished game that was shaped around the realities of therapy instead of only entertainment. That is one of the most important accomplishments in the whole project, because it shows the software design was able to adapt when the first idea was not the right fit.

6. Data Logging, User Interface, and Usability

The project also moved beyond raw interaction and into session tracking. Data logging was added as one of the code milestones. The earlier master document already planned for therapy data to be written into a text file and exported to an external USB drive, and the later code work continued in that

direction. This matters because the project is not only about making a child play a game. It also has to keep useful treatment information.

The interface work was also more than visual polish. Because the screen is small, the menu and layout had to be kept readable and usable with touch input. That included making sure game objects fit inside the display, keeping the important stats visible, and making the device open into the menu after boot. These were simple changes in description, but they were big in actual use because they made the system feel more complete and practical.

A settings and calibration area was also part of the later development. This is useful because it gives one place to review or adjust behavior instead of changing everything inside the code each time. For a future real device, this kind of separation between gameplay and system settings will make maintenance easier.

7. Main Problems Encountered and How They Were Solved

Mismatch between original game and therapy need: The first game direction was fun in theory, but it pushed short repeated breaths. The project solved this by changing to a car game that rewards steadier exhalation.

Small screen limitations: On the 3.5-inch display, some objects and boundaries were not showing correctly. The layout and game display were adjusted for the actual device screen instead of a larger monitor.

Sensor signal interpretation: The raw pulse signal needed a driver, smoothing, validation view, and calibration. Without that, the game would have reacted in a random or laggy way.

Connecting physics to gameplay: It was not enough to say the sensor works. The pulse frequency had to be converted into flow values that made sense for game control. The first calibration formula provided that bridge.

Making the Pi behave like a product: A therapy device should not require terminal commands every time. The menu-after-boot behavior helped make the Pi feel like a dedicated embedded system.

Another lesson from the overall project was that hardware, software, and physical design all affected one another. For example, when the housing changed, it changed how parts fit and how the screen area looked. When therapy logic changed, it changed what kind of game made sense. So the work was not linear. It needed several returns to earlier decisions.

8. Current Status and Recommended Next Steps

At the present stage, the strongest completed outcomes are these: the Raspberry Pi and screen work, the flow sensor is wired and readable, the first calibration relation has been created and is still used in code, the game has been finished, and the Pi can open the menu on boot. This means the project now has a working path from physical blowing, to sensor pulses, to calibrated flow estimate, to real game response.

The next work that would be most useful is formal validation rather than more feature growth. The first recommendation is to compare the current device against a trusted spirometry reference or a better controlled airflow source over several repeated trials. The second recommendation is to review the

logged session data and make sure the saved information is enough for clinical interpretation. The third recommendation is to test the system with real users or simulated therapy sessions to see whether the on-screen feedback is easy to understand and whether the target zones feel appropriate.

It would also be useful to document the final game behavior and calibration settings in one place, so later revisions do not accidentally change the physical meaning of the flow values. Battery endurance and long-session stability should also be checked, especially if the device is expected to run repeated therapy sessions without recharge.

9. Conclusion

In summary, the spirometer project made solid progress from concept to working prototype behavior. The project did not stay only at the level of design ideas. It moved through real hardware setup, signal reading, smoothing, validation, calibration, interface design, and finished gameplay. The most important technical achievement was probably not one single line of code, but the fact that the project connected physical airflow to a usable therapy game in a way that made more sense over time.

The project also showed good engineering judgment because it was willing to change course. When the original game style did not match therapy well, it was changed. When the small screen created problems, the display logic was corrected. When the game was finished, calibration was checked again instead of just assuming it was still fine. That makes the final work more reliable and more honest.

Overall, the project now stands as a strong foundation for final validation and future refinement. It already demonstrates the core idea successfully: a child-friendly respiratory therapy system where real blowing controls a game through a calibrated sensor pipeline.

Appendix A. Brief Timeline Notes

The timeline checkpoint marked settings and calibration at 50% on March 25, 2026. After that checkpoint, the game was finished and calibration was revisited again to double check that the same first calibration formula was still being used by the software. This report reflects that later state, not only the earlier timeline snapshot.