**Bubbles Summary:**

**Problem Statement:**

The inspiration for Bubbles came from seeing him as a toy dog that had a “walking the dog” feature. After an unfortunate death due to corrosive alkaline battery leakage, he was destined for the garbage. However, his fate was changed by the author who skinned Bubbles of his faux fur coat and removed the alkaline blight. In his new life, Bubbles is to be a robotic pet with autonomous behavior in addition to manual control sent over a wireless network.

**Design:**

Bubbles consists of three intelligence layers as depicted in the image below. The architecture loosely follows the subsumption architecture popularized by Rodney Brookes with each intelligence layer controlling the lower layers in a horse and rider fashion.

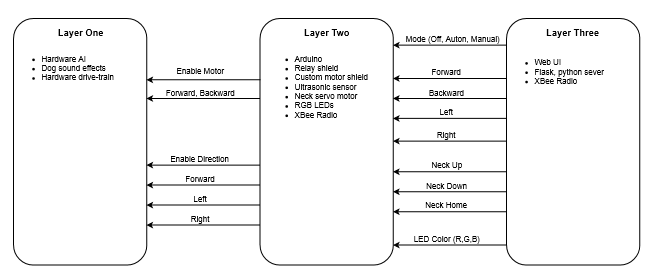


Figure 1: Horse and rider architecture for Bubbles’ design.

Layer one consists of the physical construction, mechanical systems, and built-in electronics of Bubbles from when he was a toy. Some changes were made to this layer to better control the behavior of Bubbles but the overall goal is to let the low level behavior guide the body. This layer will accept commands that follow the signals used to control the “walking the dog” remote control. One specific change to this layer from the original factory version is the exposure of the drive motor control signals to better control movement of Bubbles.

Layer two is constructed with an Arduino microcontroller, custom drive motor controller shield, Seeedstudio ultrasonic sensor, Seeedstudio relay shield, generic RGB LED ring, generic micro servo and an XBee S1 radio as shown in the schematic.

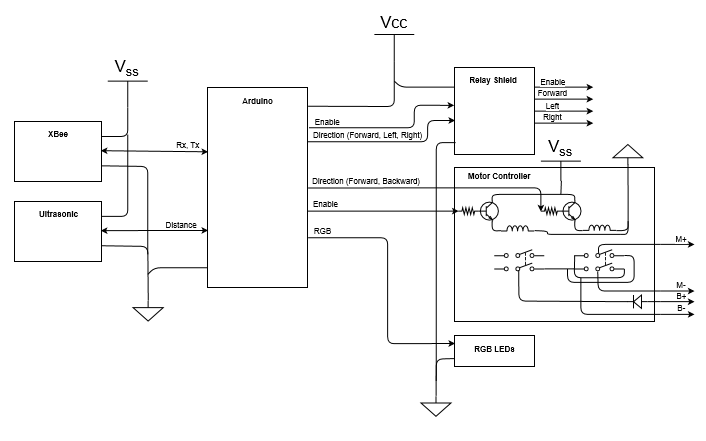


Figure 2: Schematic for layer two with detailed custom motor control board.

The connections to the first layer allow for control of Bubbles through the Arduino to run the autonomous algorithm developed, see autonomous tab for more details. Physically, the Arduino and shields are attached to the posterior of Bubbles. The XBee radio is attached to the relay shield and interfaced with software serial in the firmware. The ultrasonic sensor is attached to the lower front side of Bubbles to detect objects blocking his path. LEDs are placed into Bubbles’ head to produce a glow effect throughout the translucent plastic.

Layer three is a PC interface using the XBee radio for wireless communication. The control application exposes the option to turn off Bubbles’ second layer, thereby allowing the hardware to take control of the body or to enable layer two in either manual or autonomous mode. Manual mode allows for wirelessly moving Bubbles through finite commands, such as forward, backward, left, and right. Additional controls include RGB LED settings and head position. Software for this layer is a Flask application written in python for the server with traditional web languages for the UI.

**Future Plans and Bug Fixes:**

Several problems were encountered in the long road to realize the goal of a robotic pet with autonomous and manual control, some of these problems are slated for work as future development. First, the servo motor is dislodged from its mount causing the servo to stall and thus the system to brown out when attempting to move the servo. Currently the servo is disabled in both hardware and firmware to prevent the brown out but future work to remount the servo is needed. Additionally, moving the servo power from the microcontroller power to the motor controller power would prevent this problem from occurring in case of another hardware failure. Second, Bubbles’ battery system is inefficient, expensive, and ad hoc. Currently 4 C-cell alkaline batteries are used to power the first layer circuitry. The run time is approximately an elapsed 30 minutes. Thus for continuous use of Bubbles a large stock of batteries is need. Layer two power consists of a single 9V alkaline battery which must power the Arduino, relay shield, servo motor, RGB LEDs and ultrasonic sensor. While the battery does last longer than the 4 C-batteries, when the servo is disabled, it proves to be in adequate for continuous operation. Thus the proposed future solution is the replace the two battery system with a single 7.2V rechargeable battery with a 6V regulator for Bubbles’ hardware. While run time may be degraded, the overall life of the battery will overcome the cost of alkaline battery purchase. Finally, a problem exists in the horse and rider approach to layer control. Specifically, the problem of Bubbles’ hardware ignoring movement commands after a period of time. This problem appears to be a protective feature of Bubbles’ as toy so that a child cannot forget about Bubbles and waste battery life. Currently, the approach is the refresh the control signals to Bubbles periodically by disabling any active relays for a short duration of time. A future goal is to tune this process, along with power cycling Bubbles hardware if he becomes completely unresponsive.

**Conclusion:**

Bubbles has achieved minimum proficiency as a robotic pet with autonomous and manual control modes. Through the use of a horse and rider design methodology the different intelligence layers can express various behaviors to make Bubbles an active agent in the world. However as with any good project, there exists multiple future objectives to complete. Namely fixing the neck servo, changing over power supplies, and tuning the horse and rider control system. While not complete Bubbles has come a long way from the dead toy stained with battery deposits.

**Algorithm**