EG 10111

Penguin Pet Project Report

Steve Hanley

Kevin Riehm

Nic Zenker

Pat Myron

Garrett Dougherty

Section 5
Group 6
October 14, 2013
Instructor: Kayla L Shaw

Introduction

On Monday September 19th, each member of our learning center section five group met with different groups of fifth grade students at I2D2 (Imagination, Innovation, Discovery, and Design) and talked with them as potential customers for a robotic pet. After a few ice-breakers, the children began to talk about what they would like to see in a robotic pet. Each of the fifth graders had different opinions about what kind of pet they would like, which resulted in very diverse ideas for potential features. Some ideas, like singing and dancing, were very reasonable requests, while others like breathing fire and flying, were simply impossible to incorporate using the LEGO NXT kit. Table 1 shows some of the animals and accompanying features suggested by the fifth graders at I2D2.

Table 1. Types of Animals Suggested and Their Features

Animal	Features
Dog	Fetches, shakes people's hands, jumps
Dragon	Breathes fire, eats only humans
Hippo	Jumps, dances, cheerleads
Cheetah	Runs really fast, does your homework for you
Penguin	Waddles, can slide on its belly, hugs you, sings and dances

Although some of the ideas were quite generic, there was a fair amount of original ones as well. In the end, the decision was made to design a pet penguin. It would be a relatively original idea while remaining achievable. The fifth graders wanted their penguin to be able to waddle forward, hug, sing and dance. One of their ideas that was impossible to incorporate was programming the robot to slide on its belly. If the robot were to slide on its belly, it would fall forward onto its belly and remain in this horizontal position until the end of the program. Since a robot which could transition from a stop state to a start state was preferable, the robot would need to remain vertical for the entirety of the program, thus eliminating the belly-sliding idea. The fifth graders agreed unanimously that the pet robot be programmed in such a way that it falls asleep and snores whenever it detects the lights have been turned off.

The final prototype, a penguin named Sidney after Sidney Crosby of the Pittsburgh Penguins, was inspired almost completely by the ideas suggested by the fifth graders. Figure 1 below shows the finished prototype of Sidney.

Figure 1: Finished Sidney Prototype

Sidney waddles by spinning alternating wheels a certain number of degrees, and, using its ultrasonic sensor, the robot turns when it detects an obstacle. When the beak of the penguin is pushed its arms move up for a hug. When the lights are turned out Sidney transitions to a sleep state and when he hears a clap or loud noise, Sidney begins to sing and dance. After running through the waddle state or sing and dance state a certain number of times, Sidney becomes hungry and must be fed (this can be done by touching Sidney's beak). If Sidney is not fed after fifteen seconds of being in the hunger state, Sidney enters hibernation, which is his stop state. Table 2 below shows Sidney's states and which sensors and motors they require.

Table 2. List of States and Their Required Sensors and Motors

State	Sensors	Motors
Hunger Cycle	Touch sensor	None
Hug	Touch sensor	1 Motor
Sleep	Light Sensor	None
Sing and Dance	Ultrasonic Sensor	All motors
Waddle while avoiding obstacles	Ultrasonic Sensor, Touch Sensor, Light Sensor, Sound Sensor	2 motors

Hardware Description

We took a number of things into account in designing the physical structure of our pet. First of all, our penguin needed to look like a penguin. This was especially important considering that the intended purpose of our NXT robot was to be an interactive toy for a fifth grader who really wanted it to look like a penguin. This involved designing and building a body, head, wings, and tail that looked as realistically penguin-like as possible. The following sections review how each part of Sidney was designed and built to reflect penguin-like qualities while being able to perform the appropriate actions required in our program. Refer to Table 3 for which motors and sensors are attached to which NXT brick ports.

The Body and Tail:

Sidney's body was comprised of two motors to control the wheels, 1 motor to control the wings, and the main NXT brick. It was constructed based upon pictures of real penguins, which are vertically oriented. To mimic this, the NXT brick was built into the body above the wheel motors, with the wing motor built in behind the brick, which produced a tall, relatively thin structure. This meant that there was a fair amount of weight on the top of the structure, without a large base to support it. To counteract this, and ensure that Sidney was stable enough to move and function without falling over, two small wheels were added next to a wide, fairly realistic looking tail, in order to give Sidney four points of contact with the ground at all time. The wide tail, apart from making Sidney look more realistic, served as a final safety catch to prevent Sidney from falling over in case any of the four wheels failed for any reason. The wheel motors, representing Sidney's legs, were built in backwards, so that the bulge of the motor would not stick awkwardly out of Sidney's front. Since Sidney moved by rotating these wheels, he had to be programed with the motors going in the opposite direction of what they normally would. For instance, in order for Sidney to move forward, the motors turned in reverse. The motor on Sidney's back was used to control moving the wings up and down for the penguin's 'hug' state. Finally, we attached the sound and light sensors to the top of the body facing upwards. While they stuck out of the body somewhat unnaturally, they were optimally positioned to receive sound and light input.

The Wings:

The wings were attached to the motor on Sidney's back, and were simply constructed to look as realistically like wings as was reasonably possible.

The Head:

Sidney’s head was built to include two sensors: a sonar sensor and a touch sensor. The touch sensor was included in the head so that when Sidney was hungry, someone could ‘feed’ him by touching his ‘beak’, which in reality was the touch sensor. The sonar sensor was included in the head for two reasons. The first was for the appearance; the sonar sensor looks like a pair of eyes, so it seemed most realistic to build it into the head. The second was for a more practical reason; the sonar sensor was used to make sure that Sidney did not run into anything. By having it built into the head at a slightly downward facing angle, it was able to sense the distance to—and so help Sidney avoid—any obstacle in his path that got too close, so long as it was no lower than height of the head.

Table 3. Motors/Sensors and Their Corresponding Ports

Motor / Sensor	NXT Brick Port
Right Wheel Motor	Port A
Wing Motor	Port B
Left Wheel Motor	Port C
Touch Sensor	Port 1
Sound Sensor	Port 2
Light Sensor	Port 3
Sonar Sensor	Port 4

Software Description:

The pet’s behavior is modeled as a state machine: each feature corresponds directly to one state, and so switching between features corresponds to state transitions. This enables consistent and predictable behavior based on a well-defined structure, as well as simple translation from planning to implementation.

As shown in the state machine diagram in Figure 2, the pet has six distinct states. The start and stop conditions are combined into a single hibernation-like state in order to allow repeated use of the pet even after long periods of inactivity. When activated (by means of the prominent orange button on the front of the NXT brick), the active, sensor-based logic takes over. Programmed behaviors include a waddle state, which includes a basic form of obstacle avoidance based on the ultrasonic distance sensor; a sing-and-dance state, wherein the pet spins and loops a small clip of “Get Ready for This” by 2 Unlimited; a hugging state, in which the wings rise up and hug anyone who touches the touch sensor on the beak; a sleeping state, in which the pet “snores”, triggered from any other non-hibernation state by ambient darkness; and a hunger state, which is triggered after a certain amount of activity in the waddling or dancing state.

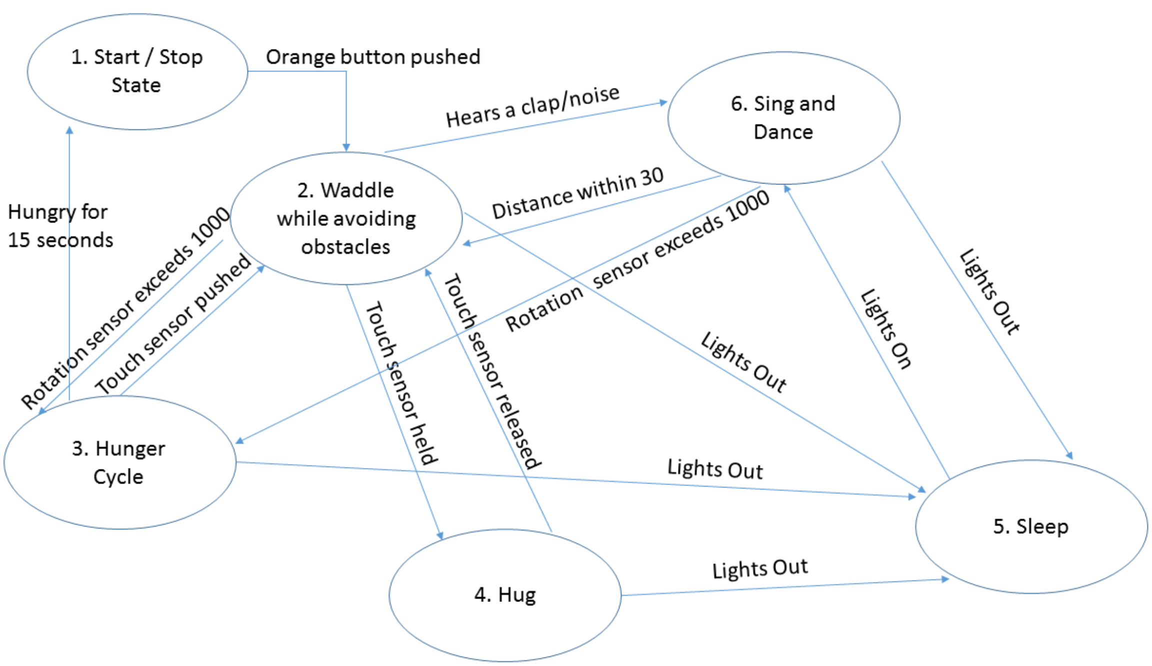

Figure 2: Penguin Robot State Diagram

Switching between states occurs as the results of comparisons with and computations on inputs from several sensors: ultrasonic (distance), sound, light, touch, motor encoders, and timers. A loop causes the behavior within each state to proceed until the state's logic determines, based on the aforementioned sensory input, to transition to another state. For example, in the sleep state, the "snoring" sound file is repeatedly played until the light sensor indicates a value above a specific threshold.

The most complicated state is state two, "waddle while avoiding obstacles". This state is initially entered from state one, the stop/start state, when the orange button is pressed on the pet's NXT brick. While in this state, the left and right leg motors are alternated, producing an effect reminiscent of a real penguin waddling. When an obstacle is detected (via the ultrasonic sensor), the power given to the right motor is changed to be zero, thus turning the robot to the right, avoiding the obstacle while maintaining the waddling behavior. As this movement is occurring, the pet is also repeatedly checking all of its sensors. If at any time it detects a sufficiently loud noise from its sound sensor, it transitions to state six, "sing and dance"; the rationale for this behavior is that loud noises are interpreted as evidence of social behavior, which the pet then desires to participate in. If the touch sensor indicates that it is pressed, the pet transitions to state four, "hug"; a press on the touch sensor indicates the close proximity of the user, and an appropriate expression of affection for the pet's owner is a hug. If the light sensor, which is set up to measure ambient light, detects a low light level, the pet transitions into state five, "sleep"; because penguins are diurnal (active during the day), a low light level would naturally trigger sleep. In addition to these instantaneous sensor-based transitions, while in the waddling state, the pet accumulates the amount of waddling it has done (this value is also added during the "sing and dance" state). When this sum exceeds a particular value, the pet transitions into state three, "hunger"; because waddling and dancing are physical activities, a pet performing them would naturally be expected to grow tired after an amount of time, requiring food - and so energy - to continue play. All of the behaviors and transitions are based on making the pet a believable model of a real penguin and the behaviors it would exhibit while waddling, with appropriate expansions (e.g., beginning to sing and dance) for the sake of the customers' enjoyment.

Results and Conclusions

After tweaking the program for Sidney, the program finally worked exactly to the specifications of the state diagram. There were many tricky parts to program; an example would be the hunger state, because the robot had to calculate a hunger value during the 'waddle' and 'sing and dance' parts of the program. Since the programming was well tested for any flaws that might arise during the demonstration, this resulted in a near perfect performance by Sidney. The only problem that occurred during the performance was the fact that Sidney got hungry during the wrong part of the demonstration. This occurred because Sidney continually gets hungry during the 'waddle' and 'sing and dance' states, therefore while these states were being demonstrated for the whole class to see, Sidney's activity was constantly adding to his hunger value. This flaw, however, was not due to any mistakes in programming, but to an incorrectly planned path created for Sidney to follow during the demonstration. The path involved a large gap between displaying/resetting the hunger cycle and showing the other states all the way to the end state. During this gap, the penguin accumulated enough "hunger" to become hungry and, therefore, did not exactly follow the demonstration pattern exactly. Other than this flaw, the demonstration proceeded exactly as planned.

There are a few aspects that could have been improved in both the demonstration and the programming. First, the transfer from the 'sing and dance' state and the 'waddle' state could have been better. The transfer was cued by the sonar sensor calculating a value less than thirty centimeters, but the sonar sensor only calculated that sensor value in the split second between repeating the music from the sing and dance state. This transfer was difficult to time and therefore did not work sometimes. This transfer could have been improved with more time and knowledge of the LabVIEW system to monitor the distance value during the entire musical clip, eliminating the timing in the transfer. Second, the path of the presentation could have been improved to accommodate for the hunger cycle changing. The hunger state could have been demonstrated more towards the middle of the presentation, so that it did not change out of place and interrupt the rest of the presentation. This would have meant fewer states after showing the hunger cycle, and therefore Sidney would not have gotten “hungry” during the latter part or the beginning part of the demonstration. Another option is that the hunger value could have been increased so that Sidney did not get hungry so quickly. These are two aspects of the project that could have been improved.

Appendix

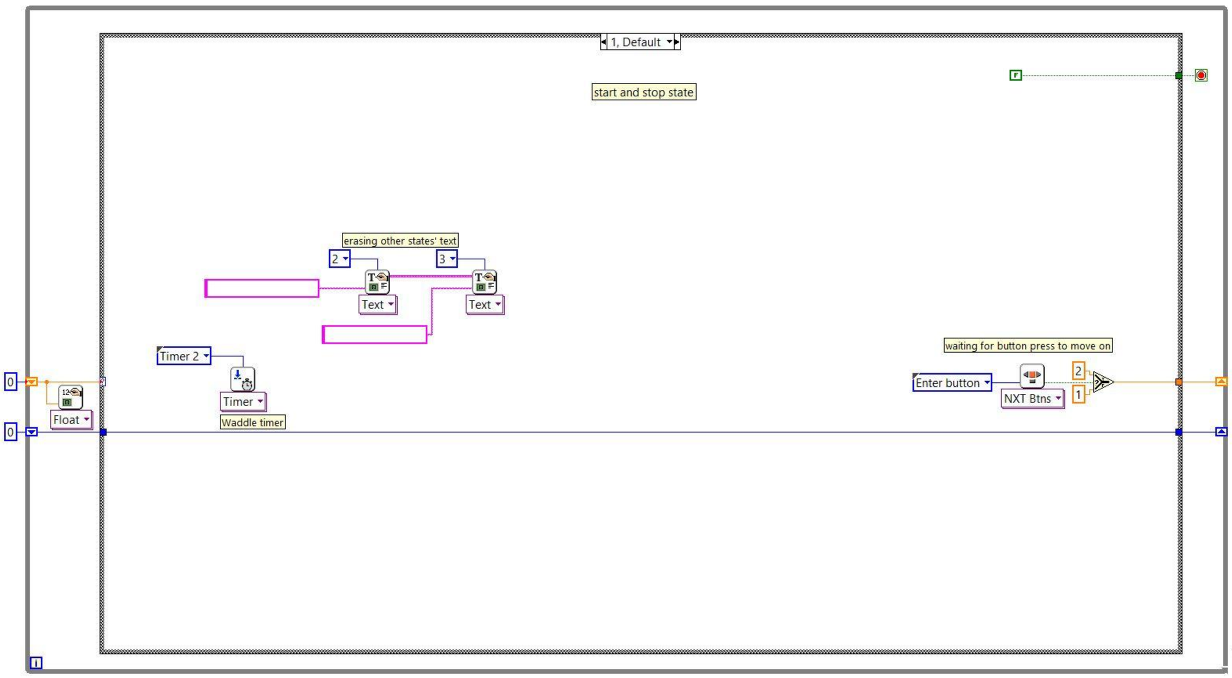

Figure 3. The Default State (also the Stop State)

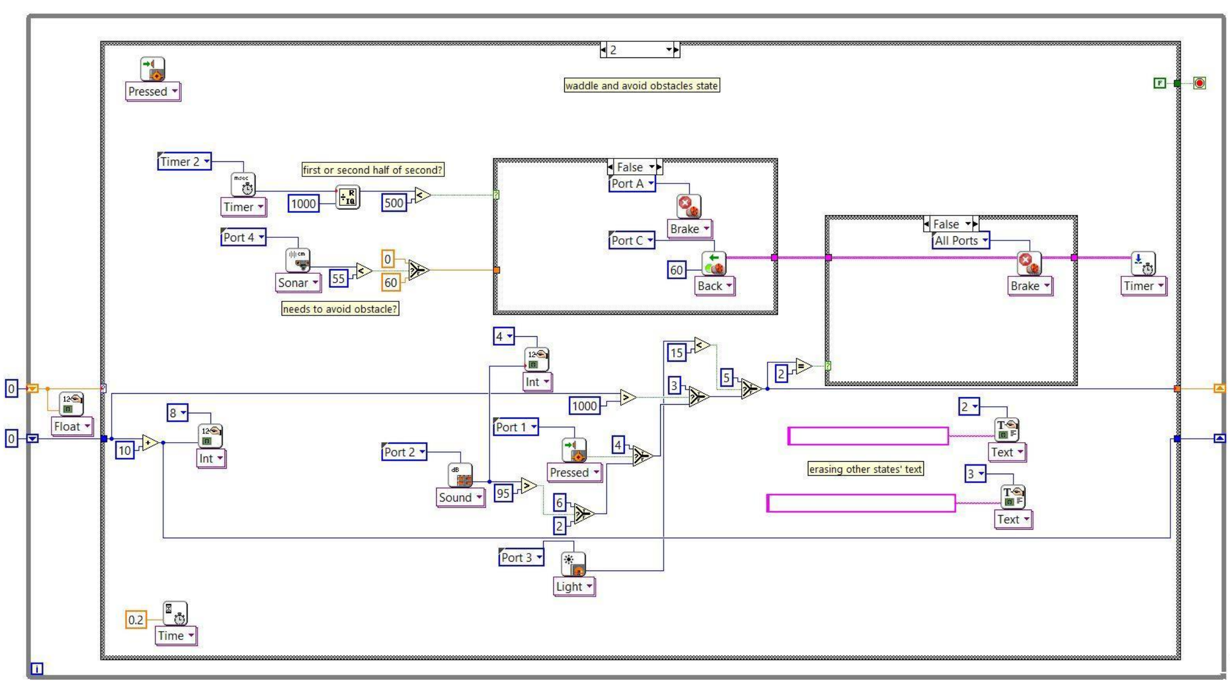

Figure 4. State 2, the Waddle State, with True Parts of Case Structures Shown

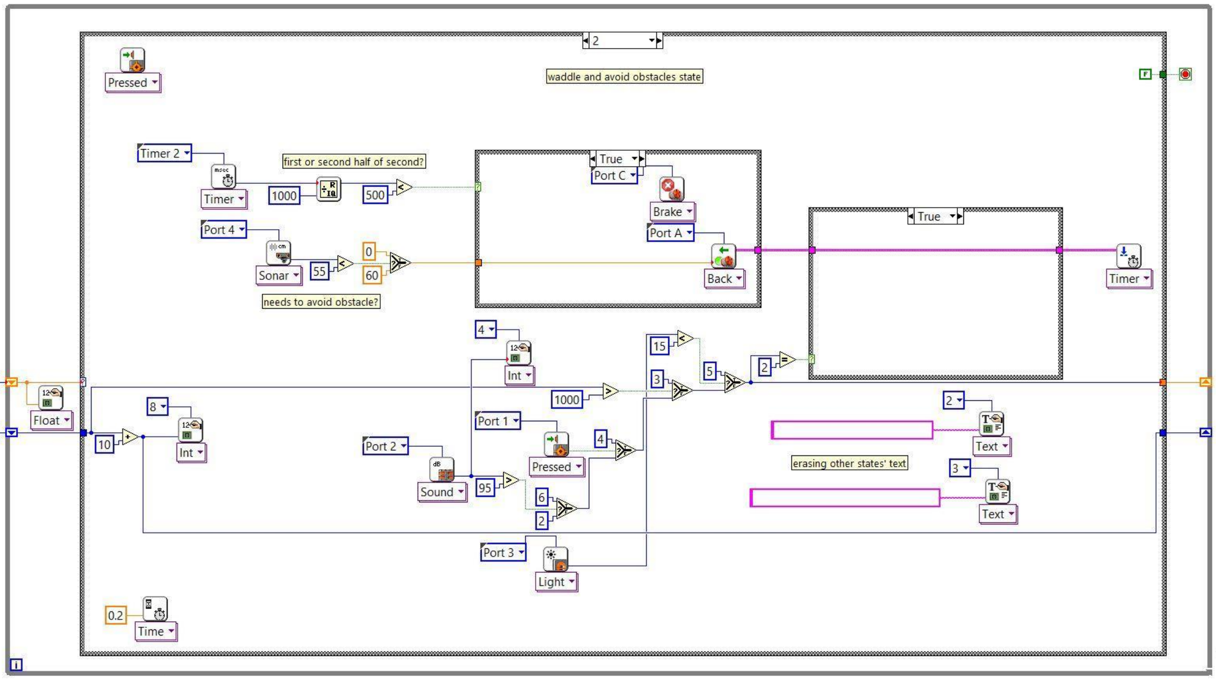

Figure 5. State 2, the Waddle State, with False Parts of Case Structures Shown

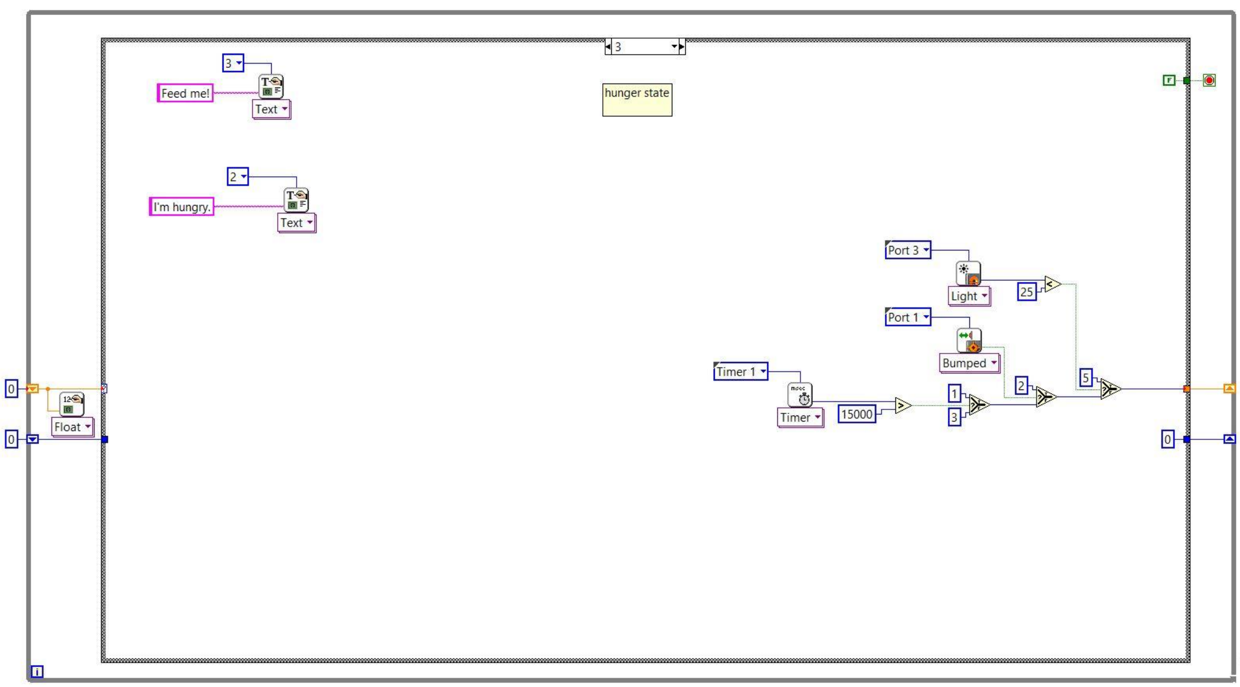

Figure 6. State 3, the Hunger State

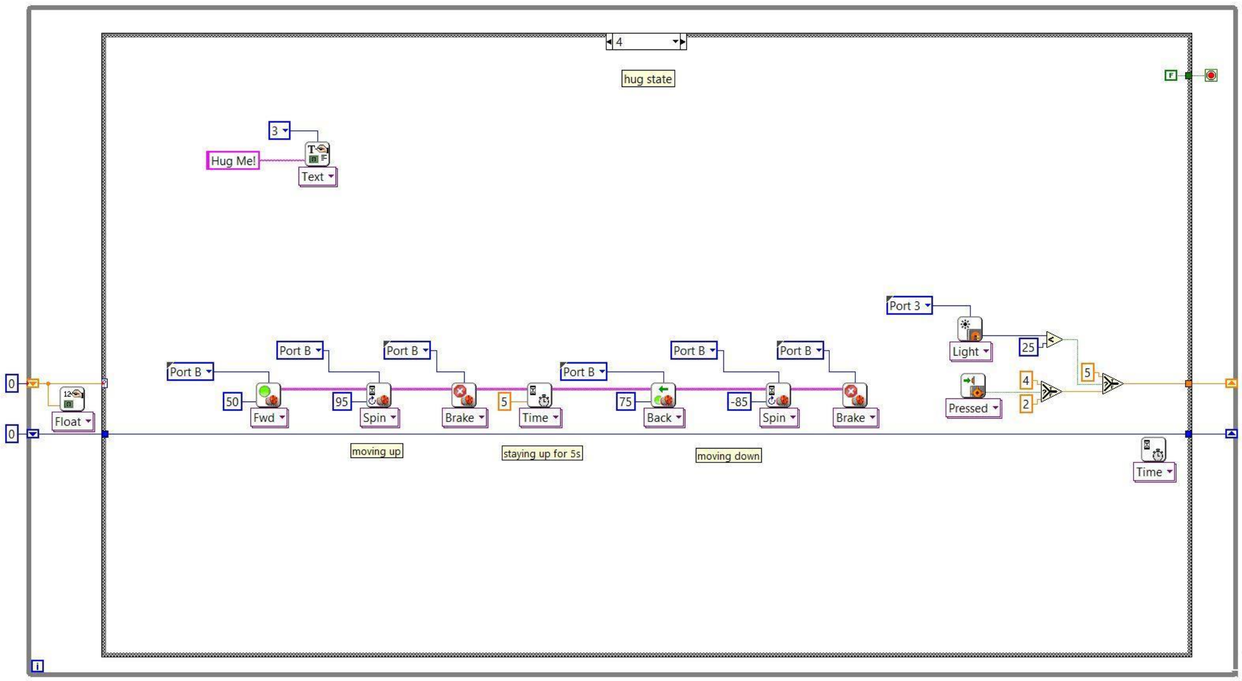

Figure 7. State 4, the Hugging State

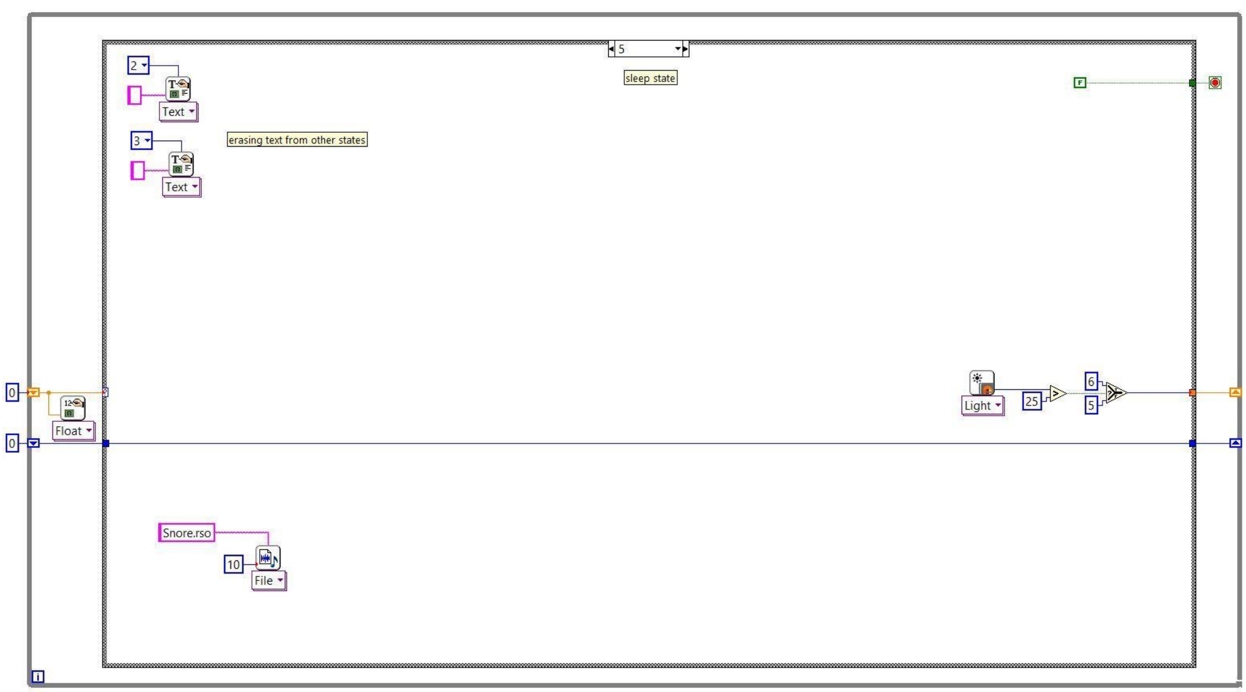

Figure 8. State 5, the Sleeping State

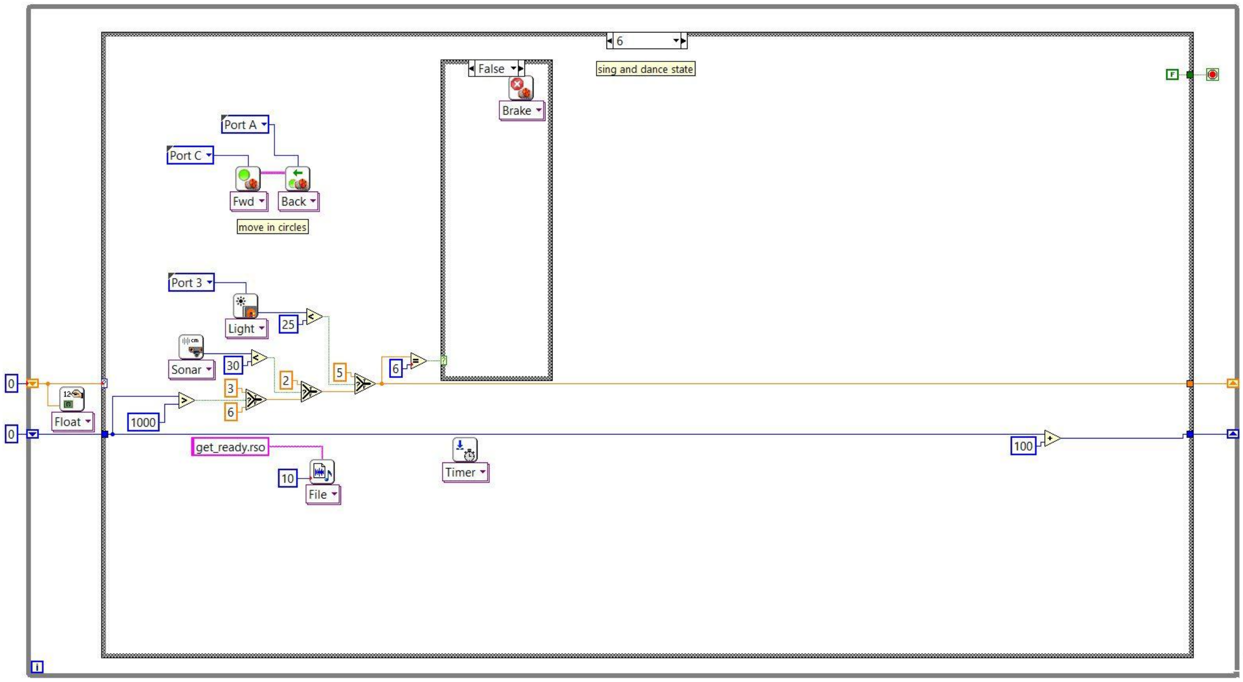

Figure 9. State 6, the Sing and Dance State, with False Case Structure Shown

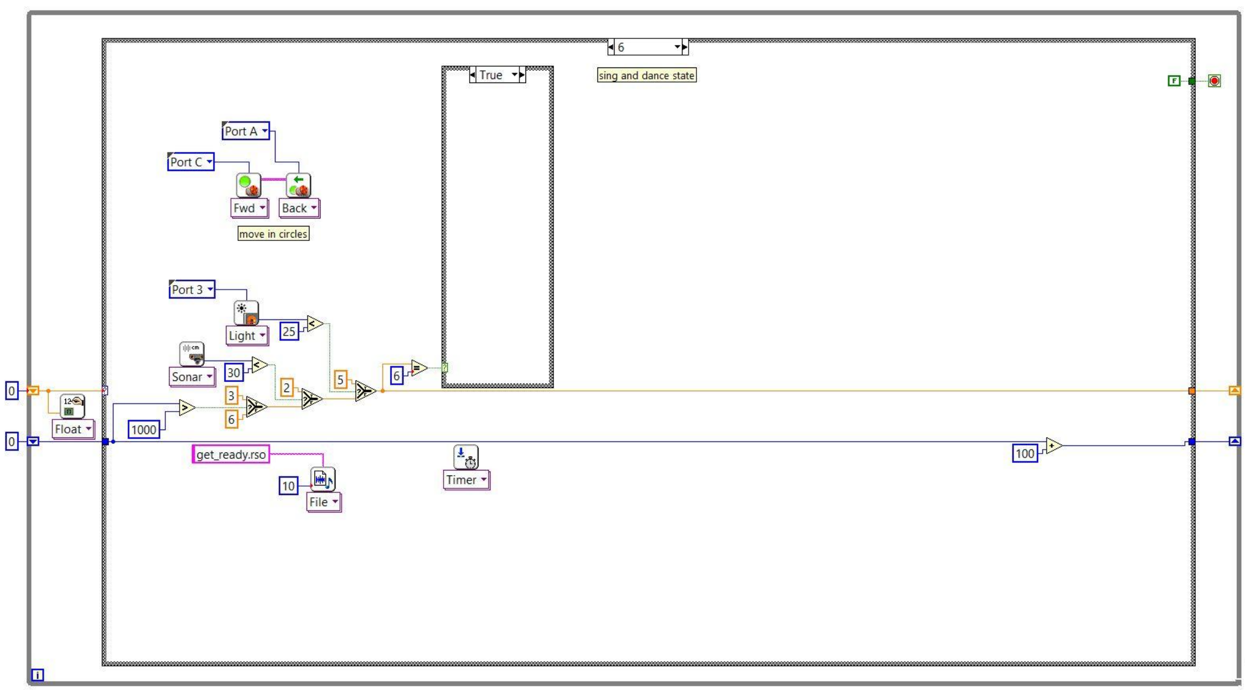

Figure 10. State 6, the Sing and Dance State, with True Case Structure Shown