Mini-Sumo Robot

J. Chang, E. Li, L. Ma, J. Sparrow, L. Richard

***Abstract* – This report outlines the design of an autonomous mini-sumo robot. The robot relies solely on infrared, ultrasonic, encoder and line detector sensors to operate autonomously within a 77cm diameter ring. The robot weights under 500g and has a 10x10cm base. The mini-sumo robot successfully executes offensive and defensive tactics on an opponent.**

I. Introduction

The purpose of this project is to construct and program an autonomous mini-sumo robot for Eleven Engineering. The goal of the robot is to push its' opponent out of the ring, just like in a real sumo wrestling match. The robot must be able to compete with other mini-sumo robots within a ring where the objective is to push the opponent out of the ring. The robot must comply with the rules and regulations outlined by the Western Canadian Robot Games (WCRG). These rules state that the robot must weight less than 500 grams and have base dimensions within 10 x 10 cm; no restrictions exist for the height of the robot. The match takes place inside a circular ring; the competing area is black and is delimited by a white ring. Various sensors on he robot are used to keep it within the bounds of the ring, to detect opponents and to trigger appropriate actions to push the opponent out of the ring.

II. Design Implementation

In order to implement our design, four subsystems were chosen: power subsystem, sensors subsystem, motor subsystem and the microcontroller subsystem. The mini-sumo robot block diagram is shown in Fig. 1. The power subsystem will provide power to our Microcontroller and all the sensors used by our robot. The sensors will gather data both as analog and digital signals and transmit it to the microcontroller. The microcontroller will use the data to trigger pre-set manoeuvres in the motors.

A. XInC2 Microcontroller

The XInC2 Microcontroller is developed by Eleven Engineering. The Mini-Development XInC2 board used in this project is a derivative of their regular board. This board is ideal for our project because of its size, the 2.8 x 6 cm board fits nicely within our 10 x 10 cm frame. Another important attribute of the XInC2 board is its multi-threaded architecture. The controller has eight threads which run simultaneously which will allow us to assign separate threads to separate subsystems.

The XinC2 Mini-DevBoard houses the XInC2 microcontroller which will be used to control the robot. This board has three sets of headers to interface with the other two PCB boards. The Mini-DevBoard allows interfacing with 41 general purpose input/output pins on seven different ports of the XInC2 microcontroller. An onboard boost converter supplies the voltage required to run the XInC2 as well as a regulated voltage for external circuitry.

Each of the eight threads behave as individual processors and have access to the main memory and the peripheral bus. With XInC2, the disadvantages of serial interrupt-based processors such as context swapping, task scheduling, unpredictable execution times, and real time operating system (RTOS) overheads are avoided. The eight processors share hardware resources with the exception of each thread's dedicated register set. Thus for the hardware cost of one processor XInC2 provides eight.

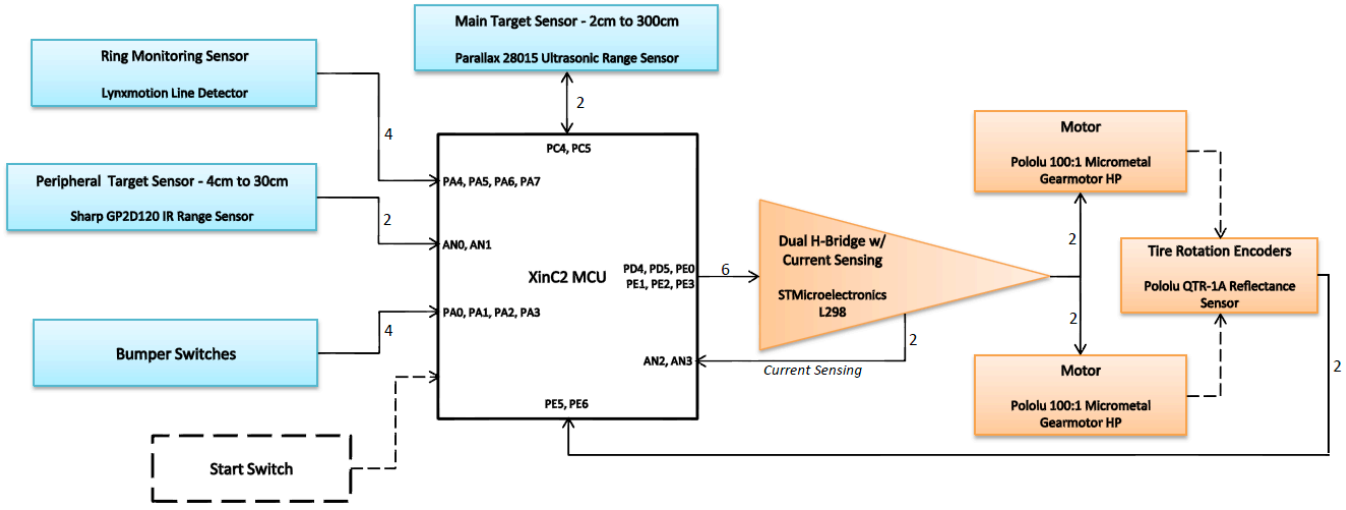

Fig. 1. Mini-Sumo Robot Block Diagram

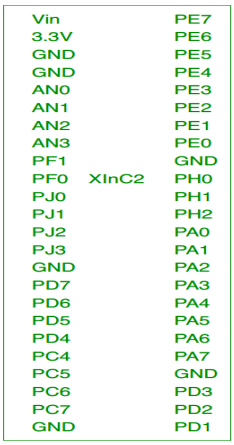

Fig. 2. XinC2 Mini-DevBoard Layout

TABLE I
XINC2 PIN SUMMARY

Pin/PCB Label	*Pin Type*	*Description*
AN0, AN1	Analog	IR sensors
AN2, AN3	Analog	Current Sensing
PA0, PA1, PA2, PA3	Digital	Bumper Switches
PA4, PA5, PA6, PA7	Digital	Line Detector sensors
PC4, PC5	Digital	Ultrasonic sensors
PD4, PD5	Digital	Motor Pulse Width Modulation
PE0, PE1, PE2, PE3	Digital	Motor Direction
PH1, PH2	Digital	Tire Encoders

Each hardware thread is scheduled to execute at 1/8 of the system clock, thus removing the overhead of an RTOS. Firmware can be written as eight independent programs, with each program running on its own thread processor [1]. We can take advantage of this architecture by assigning one thread processor to control a specific subsystem.

B. Line Detectors

The line detectors are the most important defensive sensor on the robot because these sensors will provide the robot with a sense of direction. If the line detectors are over a black surface, a logic 0 is sent and the robot will seek out the opponent. When a white surface is detected the robot is near the edge of the ring, a logic 1 is sent and the robot will switch into defensive mode to get away from the edge. The line detectors were placed at the four corners of the robot to ensure the fastest response time. States were chosen to detect a change in the state of one or two line detectors. Three line detectors were left out because three line detectors will not go off simultaneously. The line detector sensors are powered by a 5V DC signal. The output signals from the line detectors are sent to the XInC2; a black surface is a logic 0 and a white surface is a logic 1. The signal is sent to the XInC2 General Purpose Input Output (GPIO) Port A. The maximum threshold that the GPIO's can handle is 3.3 V. The signal coming out of the line detector is sent to a voltage divider and then signal from the midpoint of the voltage divider is then sent to the XInC2 port. This effectively divides the signal in half, thus allowing it to be under 3.3V. The voltage divider we chose is a 50/50 divider. Two resistors of 270kΩ were chosen in order to neglect the internal resistance of the line detector.

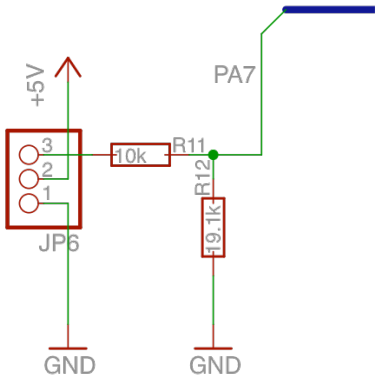

Fig. 3. Line Detector Schematic

C. Tire Encoders

The mini-sumo robot uses QTR-1A encoders in order to keep track of the distance travelled by the wheels of the robot. An encoder for each wheel is inserted into the tire. The encoders operate by reading the teeth on the inside of the tire. By tracking the voltages transitions/changes of the encoder as the teeth of the inner tire are read, the distance that each wheels travels can be determined.

As the sensors come across a strong reflective surface such as a white surface, the output tends towards 0V. As the sensors come across a weak reflective surface such as a black surface, the output voltage tends to towards the supply voltage. The voltage supplied to the encoders is 5V from the regulator. As the encoders read the teeth of the tire, the low voltage range varies from 0.246 -1.06V and the high voltage range varies from 2.08- 2.89V. The ranges of these voltages depend on how deep the encoders are embedded in the tire. Since these 'raw' voltages are adequate voltages for the digital I/O pins of the microcontroller, a voltage divider are not needed to reduce the output signal voltages of the encoders.

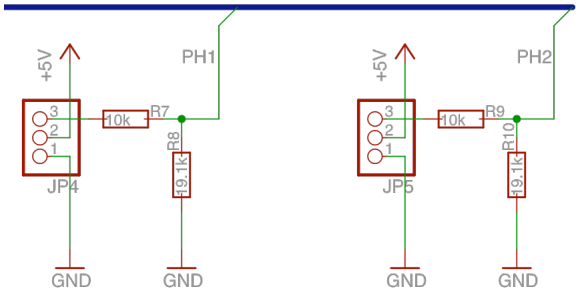

Fig. 4. Tire Encoders Schematic

D. Bumper Switches

The bumper switches are used to establish if contact has been made with the opponent. There is a bumper switch for each side of the robot for a total of four bumper switches. Each bumper switch is tied to the 3.3V rail and a 1kΩ resistor. The signal from the bumper switch is then sent to the

GPIO ports (PA3-PA0). The bumper switch is tied to 3.3V until contact is made which connects the bumper switch to ground. XInC2 interprets these signals as a logic 0 or 1.

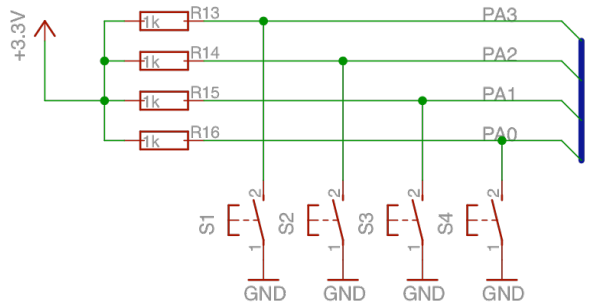

Fig. 5. Bumper Switches Schematic

E. Ultrasonic Sensor

The Ultrasonic sensor used in this project is the PING parallax sonar sensor. The schematic of the sonar is shown in Fig.6. The sonar requires two pins on the Mini-Dev board, one pin to send a pulse (PC4) and the other pin to receive (PC5) the echo pulse. The XInC2 will send a logic 0 to the 2N4401 NPN BJT base junction which will turn it off. This allows the 5V tied to the collector junction to send a trigger pulse to the sonar. In order to receive a signal, a 1 is sent, turning on the BJT. Now the anode of the diode is at a lower potential than on the cathode side, so the sonar will return the signal through the voltage divider and back into pin PC4. In order to send and receive the sonar signal on the same pin without being harmful to the GPIO ports, a BJT is used as shown on Fig.6. The voltage divider is a 50/50 divider that uses two (2) 1.5 MΩ resistors. A 4.3 kΩ resistor is used on collector junction of the BJT while a 15 kΩ is used on the signal path.

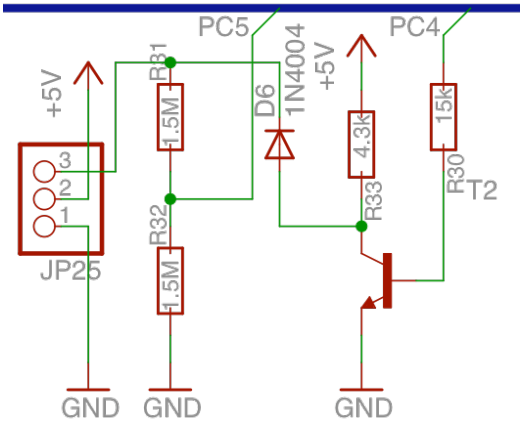

Fig. 6. Ultrasonic Sensor Schematic

F. Power Supply

The XInC2 microcontroller, sensors and motors will be powered by two 9V batteries. Two batteries are required to ensure that the subsystems can draw enough current. Having one battery dedicated to the motor systems and the other battery for the rest of the electronics also reduces noise in the system.

Regulators are used to drop the supply voltage from 9V to 5V and 3.3V. The 5V rail will power the infrared, ultrasonic, encoder and line detector sensors while the 3.3V rail will power the XinC2 microcontroller and the bumper switches.

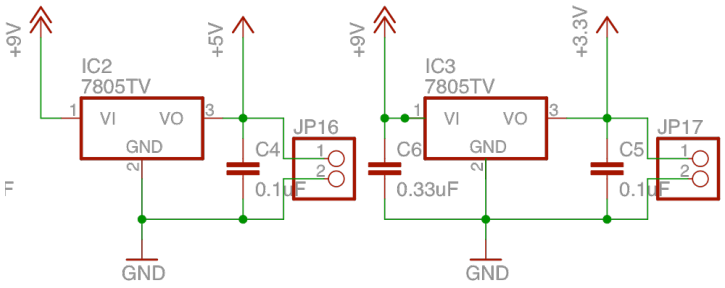

Fig. 7. Voltage Regulator Schematic

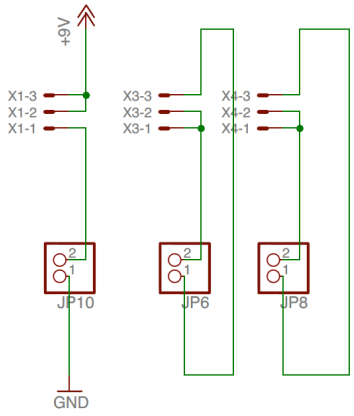

Fig. 8. Power Supply Switches Schematic

G. Infrared Sensors

The Sharp (GP2D120) IR sensors play an important part in the offensive strategy of the Mini-Sumo Robot. There are 2 IR sensors mounted on the front of the sumo robot 3 cm behind the front panel to avoid the "blind spot" limitation of the sensor. The Sharp sensor came with a datasheet that shows the relationship between voltage and distance, which was verified during component testing. The sensors operate at an input voltage of 5V and output voltages up to 3.2V, depending on the distance of the target object from the sensor. The XinC2 Microcontroller analog ports (AN0, AN1) can handle a maximum of 1.8V, so a voltage divider, consisting of a 1.5K and a 2.2K resistor as shown in Fig.9, is used to drop 3.2V to below 1.8V.

H. Motors

The motors subsystem consists of a L298 dual full-bridge driver, two 100:1 micro HP gear-motors manufactured by Pololu, and current sensing feedback for each motor. The driver allows for bi-directional operation of 2 separate motors which is necessary for our application. In order to protect the driver from destructive back current from the motors, freewheeling diodes were used to allow a safe path for the

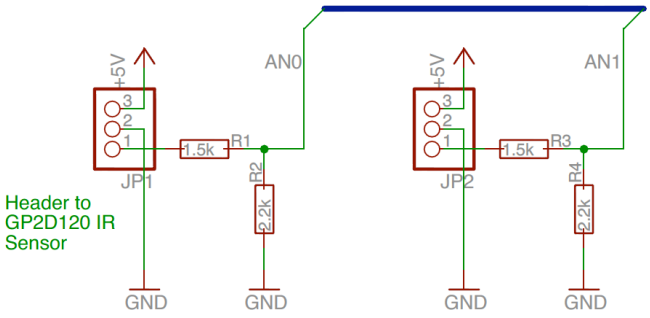

Fig. 9. Infrared Sensors Schematic

current to flow. The gear-motors have a free-run speed of 315rpm, and draws as little as 80mA when operated at a rated voltage of 6V. When the motors are stalled, each one draws a current of 1.6A which is below the driver's maximum output current of 2A. This allows the motors to achieve a lot of torque which is required to push an opponent out of the ring.

Current sensing feedback is achieved by inserting a 0.2Ω resistance, achieved by 5 1Ω resistors in parallel, between the SEN pins of the driver and ground (as shown in Fig. 11) and sensing the voltage across the resistance using one of the XInC2 ADC pins. This voltage is proportional to the current flowing through the motors and provides information on the torque of the motors. The speed of the motors can be determined using the tire encoder feedback. Together, this information allows for accurate representation of the state of the motors, such as whether they are spinning out or if traction is achieved.

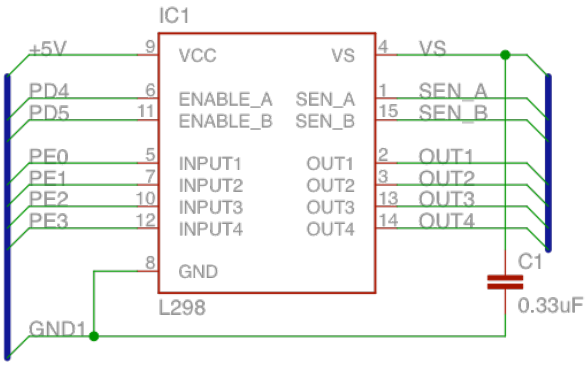

Fig. 10. L298 H-Bridge Schematic

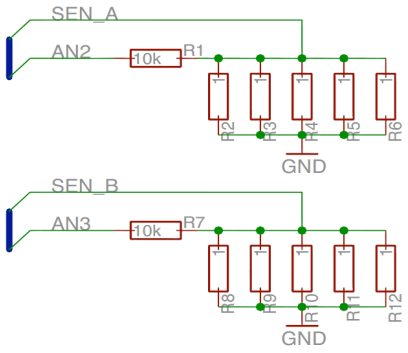

Fig. 11. L298 Current Sensing Schematic

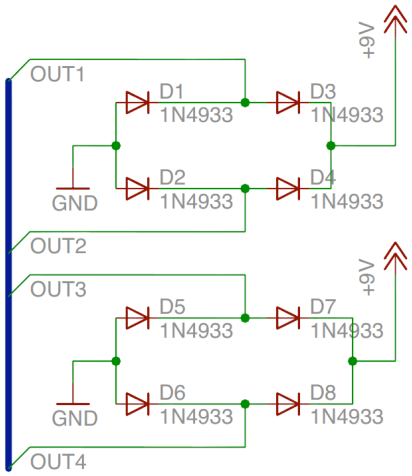

Fig. 12. Motor Freewheel Diodes Schematic

III. Software Design

The mini-sumo robot has to be completely autonomous, meaning it will have to navigate through the ring and find the opponent completely on its own. To perform these tasks reliably and efficiently, the robot must depend heavily on the sensors and firmware developed for these sensors.

A. Line Detector Software

As stated above in the hardware design section, each line detector sends a 0 V signal to the XInC2 when over a black surface and a 2.5V signal to the XInC2 when over a white surface. A 0V signal is interpreted as a logic 0, while the 2.5V signal is interpreted as a logic 1. All the information from the sensors is stored on one thread as a gatekeeper function. Made a subroutine. We have decided that there are eight combinations that need to be monitored: front left corner, front right corner, back left corner, back right corner, front side, back side, left side, and right side. Each line detector represents 1 bit in an 8-bit word. When the program detects one of the aforementioned states, the defensive mode is activated and the robot will manoeuvre towards the center of the ring. If all states are cleared, the robot will enter Hunt mode. The figure shown below illustrates the high-level programming.

B. Tire Encoder Software

The tire encoders read the transition between the tire, a black surface, and the teeth of the wheel hub, a white surface, to determine the distance travelled by the wheels. This information is used in conjunction with the current sensing to provide feedback to the microcontroller and determine if the robot is stalling or slipping. The feedback from the encoders provides the XinC2 with the ability to control the distance each wheel of the robot travels in order to perform various manoeuvres, such as 90 degree turn, wide turns or stationary spins.

The inner wheel hub consists of 12 teeth which correspond to 24 voltage transitions per rotation. With each tire having a

diameter of 4.2cm, it was calculated that a full rotation of the wheel will travel 13.195cm and that each transition corresponds to approximately 0.55cm travelled.

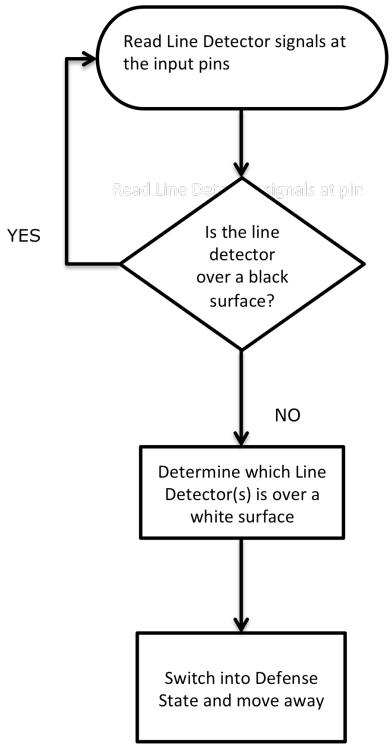

Fig. 13. Line Detector Flowchart

A subroutine program for the encoders reads the inputs from digital I/O pins, H1 and H2. The program then branches into one of four different state loops depending on the initial encoder input reading. The different states are: both encoder inputs are low, both encoder inputs are high, encoder 1 is low while encoder 2 is high, and encoder 1 is high while encoder 2 is low. The state loops continuously read the input and remain in the loop until a change in state is detected, then the code branches to one of the three counters. One counter branch increments the counter for encoder 1, another counter branch increments the counter for encoder 2, and the last counter branch increments both counters if a change in state of both pins is detected. The counters for each encoder are stored in RAM right after the counter increments.

As mentioned earlier, the encoders count the number of transitions from low to high or high to low voltages. Each transition is approximately 0.55cm travelled by the wheel. Another subroutine is made to convert the encoder counter to distances travelled. Since the XinC2 cannot multiply register values or RAM variables, a loop is needed to continuously add the counter to itself until the multiplication is complete. It should also be noted that the microcontroller cannot accept the direct multiplication of a decimal constant and will round results to the nearest whole number when using the unsigned divide subroutine. The conversion subroutine takes the encoder counter and multiplies it by 10 and then divides the result by 18 to give an approximate multiplication of 0.55. This operation is performed by adding the counter to itself 10 times, then dividing the result by 18 using the unsigned divide subroutine. The reason for multiplying by 10 and dividing by 18 rather than multiplying by 55 and dividing by a 100 is just to provide a 'safer' conversion process. This method is 'safer' because the maximum value that can be stored in a register is 2^{16} (=65,536). If the multiplication results in a value higher than this, errors in calculation will occur. Multiplication by a smaller number (10 rather than 55 in this case) will be less likely to ever reach this maximum value and cause errors. This conversion routine is utilized for both encoders and resulting distance in centimetres is stored into RAM.

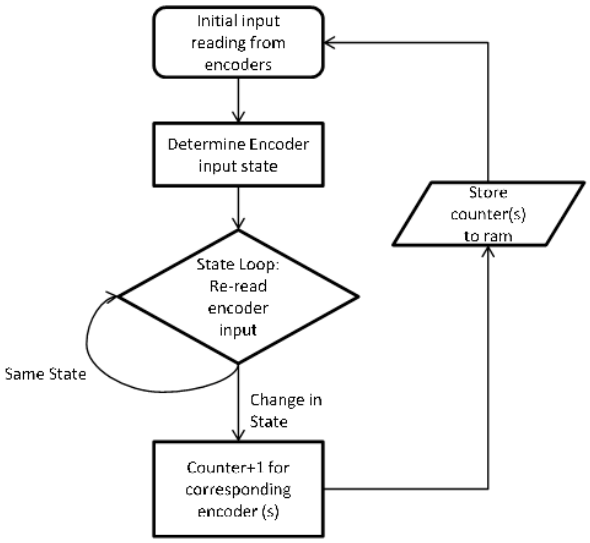

Fig. 14. Tire Encoder Flowchart

The encoder conversion subroutine is used in the counter branches of the encoder counter subroutine so that the values of distance travelled by the wheel will be updated immediately after an increment in the encoder counter. The subroutine will be run by one thread, continuously updating and storing results in RAM for access by the main program.

C. Infrared Sensor Software

The IR subsystem software begins with the initialization of the hardware by clearing all registers, stacking previous variables, and configuring the XInC2 analog-to-digital converter. Once the initializations are complete, the ADCdata is checked to determine if the information is from the proper channel. If so, then this information is stored for further processing, whereas if the information is from the wrong channel, it is disregarded. This is required to receive the appropriate information which represents the distance of the opponent. The ADCdata register is a 10bit register, so XinC2 will read out the distance as a range in hex, 0x0000 to 0x03FF. The data is then converted to a distance in centimetres. The robots reaction will depend on how close the opponent is to the Mini-Sumo Robot. One problem that was encountered was that the acquired data was unstable, typically varying by one byte value. In order to solve this problem, an average of five reading was taken before storing into RAM.

The XinC2 capabilities allow for the IR sensors to be continuously read in and the RAM registers to be updated without any major delays.

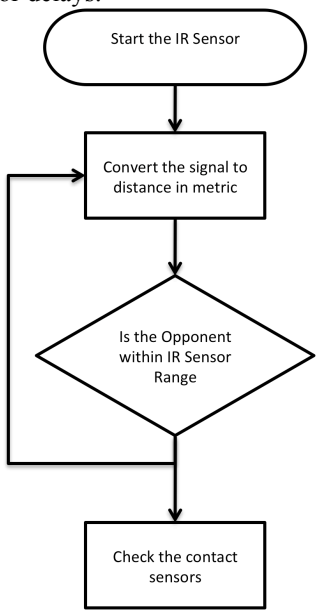

Fig. 15. Infrared Sensor Flowchart

D. Sonar Software

The sonar software triggers the sonar, and then waits for a response. This subroutine will send a 5μs trigger pulse to the sonar and then wait for 690μs before repeatedly checking for the signal line to go high. After which, the program will initialize a counter and wait for the signal to return to 0. If no signal is received, then the counter will time out after 21ms. The resulting count is multiplied by 1.2 to obtain a reading in millimetres. This data is then stored in a RAM register for use by the main program. Each increment in the counter results in one reading and the entire routine takes about 21ms to execute.

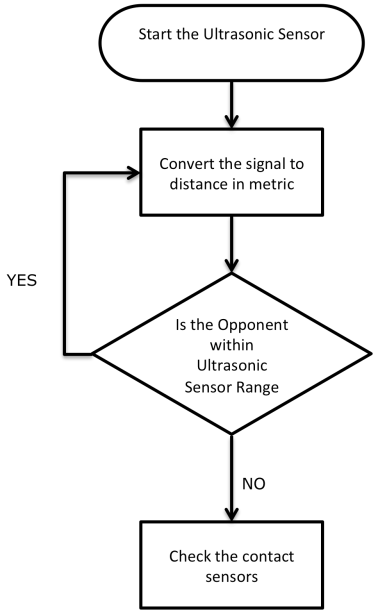

Fig. 16. Ultrasonic Sensor Flowchart

E. Motor Software

The motors are operated using a PWM signals with varying duty cycles. These signals are generated using the XInC2's timer facility which is configured with the time length of one cycle as well as the time after the start of a cycle that the signal should go low. This signal is applied to the driver on the enable pins. The average voltage sent to the motors can be controlled by changing the duty cycle of these signals. The driver has two inputs, one for each motor, to control direction. These inputs should always be opposite of each other or both off, otherwise a short is created.

F. Complete System Programming

The firmware for the entire robot takes full advantage of the eight threads of the XinC2. Threads 0, 1, and 2 are for main strategy programming. Thread 3 continuously loops the encoder subroutine, while Thread 4 continuously loops the subroutines of the bumper switches, line detectors, and encoder/infrared conversion to centimetres. Thread 5 continuously loops the subroutine for analog sensors, whereas Thread 6 continuously loops the ultrasonic sensor subroutine and ultrasonic conversion to centimetres subroutine. Thread 7 configures the ports and other resources of the XinC2, then continuously loops the subroutine controlling the motors.

TABLE X
THREAD USAGE SUMMARY

Thread #	Purpose
0	Strategy programming.
1	Strategy programming.
2	Strategy programming.
3	Updates tire encoder data.
4	Updates bumper switch, line detector, and LED data. Also converts data from encoders and IR sensors to cm.
5	Updates current sensing and IR sensing data.
6	Updates ultrasonic sensor data.
7	Configures ports on XinC2 and controls/updates motors.

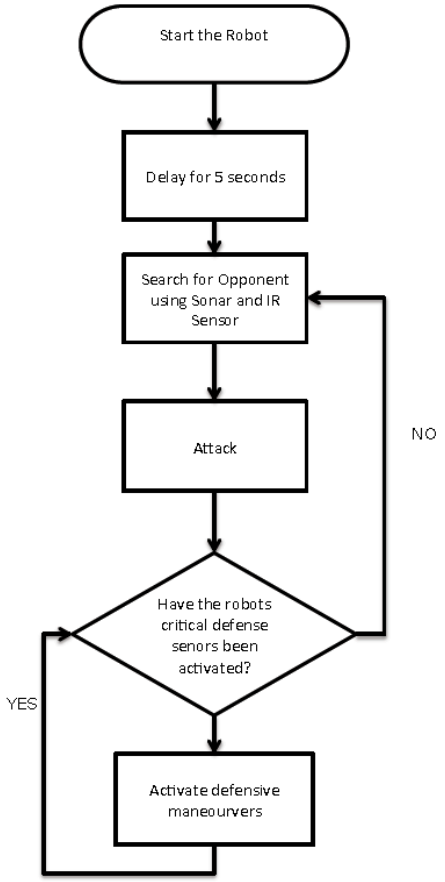

Fig. 17. Main Program Flowchart

IV. TESTING

A. Breadboarding

Initially, each subsystem was tested by constructing the circuit on a breadboard according to the schematics. This was an easy way to layout the circuit in order to test if each component was working properly, as well as the general circuitry. Components were interchanged easily if the subsystem was not performing adequately. Critical measurements and observations were taken using an oscilloscope to ensure the circuit was operating properly and power consumption was acceptable.

The breadboarding stage of testing was fairly trivial in this case because the subsystem circuitry is basic in nature and all circuits worked as expected, except the ultrasonic subsystem. It was discovered that one pin on the sonar was used to send and receive the signal, which required extra circuitry in order to separate the signal. Breadboarding of this circuit was important to obtain the lowest possible power consumption while still successfully operating the sonar. Motor testing was also performed at this stage by mounting the H-bridge and supporting circuitry on a breadboard, then operating the motors in both directions.

B. Printed Circuit Board (PCB)

The PCB's needed to be inspected after they were manufactured to ensure that each trace was where it should be and that the traces did not accidently short circuit. Some traces were connected at places where they should not have been, so they were carefully cut out. After soldering components onto the PCB, each PCB was tested individually in order to make sure that it performed identically to when it was tested on the breadboard.

The main problems encountered at this stage of testing were connected traces at places they shouldn't be and excessive solder which created short circuits between adjacent traces. The errors in traces were resolved through careful inspection of the boards and thorough comparison with the schematics. Soldering issues were dealt with by using de-soldering strips to remove excess solder. Some components were re-soldered in order to obtain good connections without unwanted short circuits. Great care was taken when soldering close connections in order to minimize the likelihood of such cases.

C. Test Program

Testing of the Mini-Sumo Robot was carried out using a test program which mimics our final program. The "Test Program" is a compilation of various test routines which can be accessed from the main menu of Eleven Engineering's XDE Development Toolkit. These routines include a RAM register dump routine, a routine that displays the status of the sensors, and a routine called "Set Drive". Set Drive allows the user to input data of a specific manoeuvre and run the "Drive" subroutine. The dump routine allows the user to get an instant update of any register defined in the "Long Data.asm" file. The sensors routine displays the data in all the RAM registers.

Besides allowing effective testing of the robot functionality, the test program is beneficial because we can simulate different situations by manually setting the sensors during an action. This is helpful because different manoeuvres and scenarios can be simulated quickly, thus showing which strategies are most beneficial for the whole system.

V. CONCLUSION

The mini-sumo robot was designed and implemented successfully according to the specifications required by the client. It met the 500g weight restriction, the 10x10cm base size restriction, and is able to manoeuvre around the ring and locate the opponent, while carrying out both defensive and offensive strategies. A diverse array of sensors is used to provide the microcontroller with as much information as possible. This information is then interpreted by various subroutines and appropriate actions are implemented. Possible improvements that could be made to the mini-sumo robot are using cheaper encoder than the Lynxmotion line detector to save on costs and using different positioning of the tire encoders to reduce fluctuations in low and high voltages.

ACKNOWLEDGMENT

A special thanks to our client, Cory Duce, of Eleven Engineering, as well as Loren Wyard-Scott and Edward Tiong.

REFERENCES

[1] P. Jacobsen, *XinC2 Users Guide Version 1.1*, published by Eleven Engineering Incorporated, 2006.
[2] D. Beutel, *XinC2 Assembler Guide v1.1*, published by Eleven Engineering Incorporated, 2007.
[3] D. Beutel, *XinC2 Development Tools v1.1*, published by Eleven Engineering Incorporated, 2007.
[4] D. Beutel, *XinC2 Libraries v1.1*, published by Eleven Engineering Incorporated, 2007.
[5] C. Duce, *XinC2 miniDev Development Board User Guide Version 1.0*, published by Eleven Engineering Incorporated, 2008.
[6] *XinC2 Programmers' Card Instruction Set Summary*, published by Eleven Engineering Incorporated, 2002.
[7] D. Beutel, *XinC2 Quickstart Project Overview*, published by Eleven Engineering Incorporated, 2007.
[8] *XPD Download Instructions v1.0*, published by Eleven Engineering Incorporated, 2007.
[9] Lynxmotion Inc, "Single Line Detector, Reflective IR Sensor," published by Lynxmotion Inc, 2006. http://www.lynxmotion.com.
[10] Polplu, "QTR-1A Reflectance Sensor," http://www.pololu.com/catalog/product/958
[11] Sharp Corporation, "Analog Output Type Distance Measuring Sensor, GP2D120XJ00F," 2005,
[12] ST Microelectronics, "Dual Full-Bridge Driver," 2000, http://www.st.com
[13] Parallax Inc, "PING))) Ultrasonic Distance Sensor (#28015)," 2008, http://www.parallax.com
[14] ST Microelectronics, "Postive Voltage Regulators," 2008,