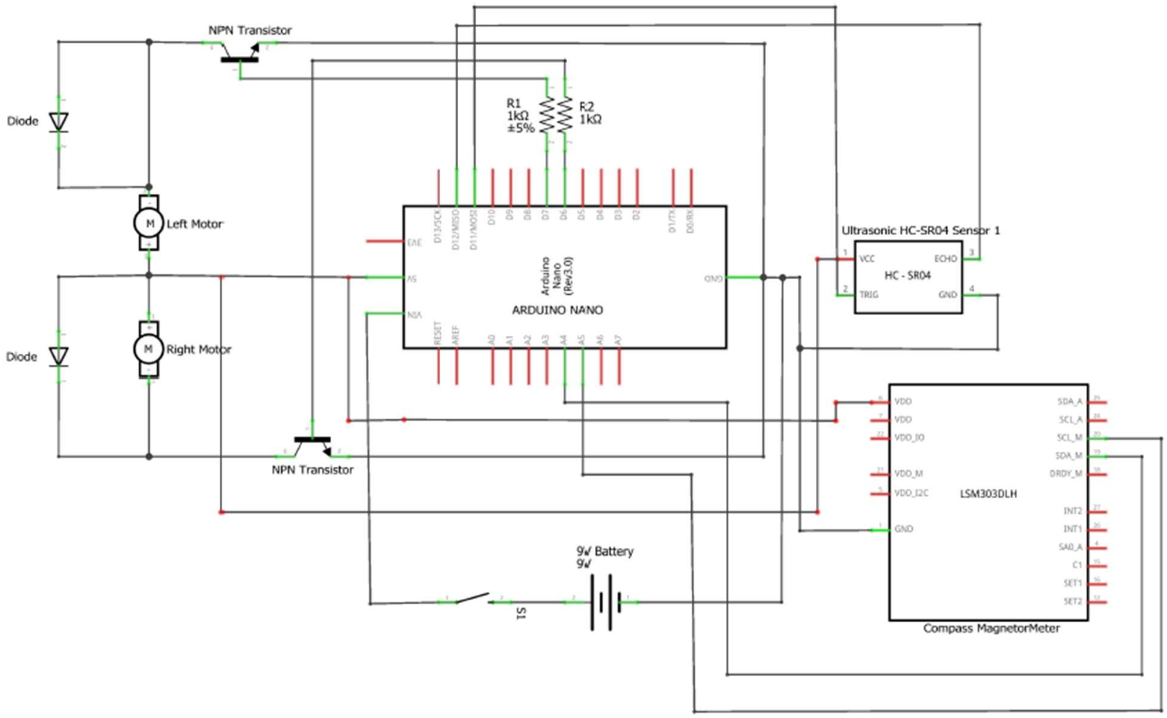
ELEC4403 Project Design Document

Task allocations

Vlad Matveev - Coding compass, coding main (full) program layout
Dickson Hee - Coding distance sensor, documentation
Daniel Santosa - Coding motors, code checking, hardware wiring

Hardware circuit diagram



Main Components :

1. Arduino NANO to control all boat functions
2. Motor to propel boat through water
3. Ultrasonic sensor used to determine distance of wall from boat to turn the boat and stop at the end.
4. Compass magnetometer used to determine boat direction to keep it cruising straight to wall and not veer off.
5. Battery, NPN transistor, resistor, etc.

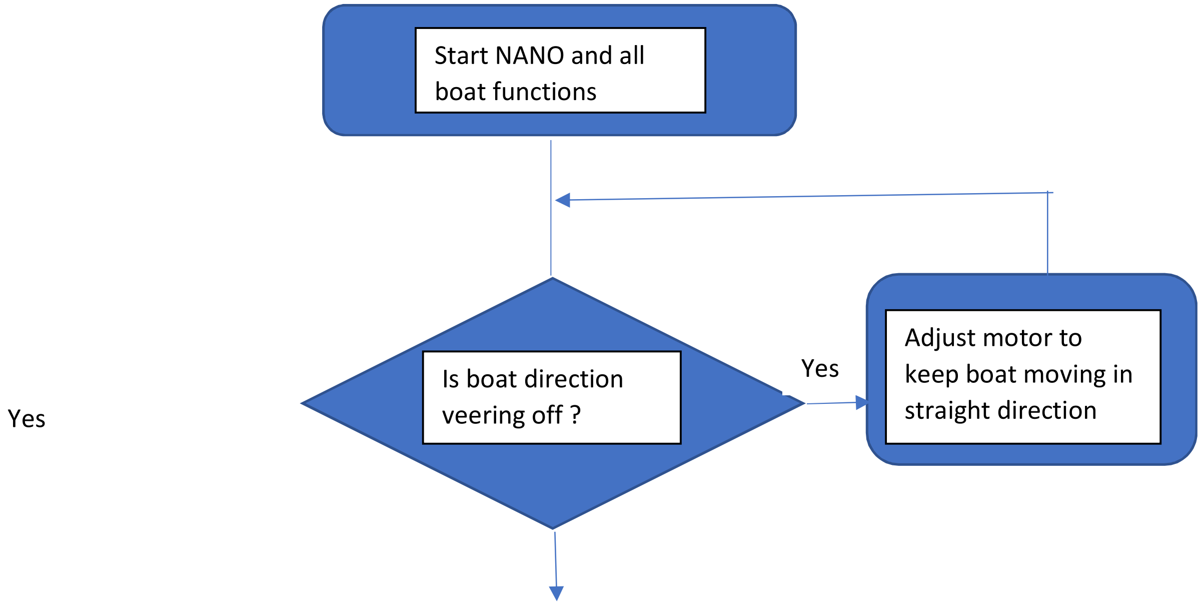
Main obstacles and issues were fitting everything inside the model boat and getting certain components to work properly e.g. compass which had lots of problems configuring to work properly and also making everything work together.

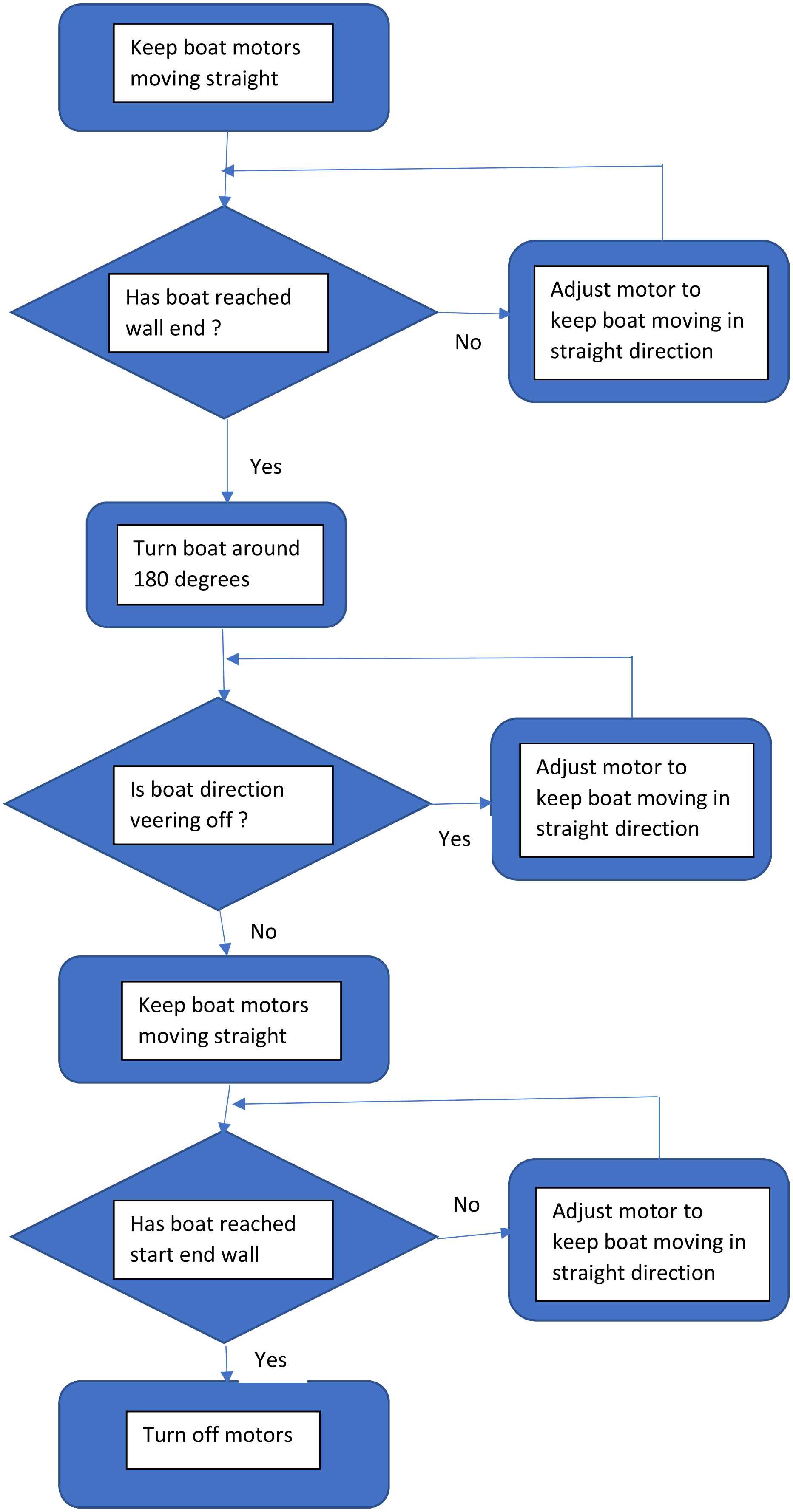
Project Budget

ELEC4403 Project Budget				
Part Name	Part Code	Quantity	Supplier	Price
Remote control model boat	C101A	2	eBay.com	$30
Ultrasonic distance sensor module	HC-SR04	1	jaycar.com.au	$15.95
Triple Axis Magnetometer HMC5883L	HMC5883L	1	adafruit.com	$13
Triple Axis Magnetometer	HMC5883L	1	core-electronics.com.au	$7.88
Arduino Nano	ATMega 328		UWA	$5
Switch PCB	SPST		jaycar.com.au	$0.95
Schottky Diode	IN5819	2	jaycar.com.au	$1.60
NPN Transistor	TIP122	2	jaycar.com.au	$4.30
			TOTAL COST	$79

Software design description & diagram

The design of the program mainly revolves around importing all the necessary libraries for the compass to work, as it is advanced enough to require an external library to fully function. Then there are a few definitions/declarations of variables that are used by each of the sensors. After that, there are a few setup functions for each of the major components: compass, distance sensor and the motors, all these do is initialise the correct pins for inputs and outputs for each respective part. The rest of the functions are self-explanatory in their naming, and are quite isolated from each other, which makes testing them much easier.
The bulk of the logic for the boat's operation is stored in the main loop() call, which uses flags to check for the current state of where the boat is in its execution, i.e. 'wall1' being the first wall that is to be/was met, wall2 being the finish line etc. A unique state of execution can also be testing by setting the flags prior to starting execution, i.e. making the boat turn without a wall nearby.


Start NANO and all boat functions
Is boat direction veering off ?
Yes
Adjust motor to keep boat moving in straight direction
Yes


Keep boat motors moving straight
Has boat reached wall end ?
No
Adjust motor to keep boat moving in straight direction
Yes
Turn boat around 180 degrees
Is boat direction veering off ?
Yes
Adjust motor to keep boat moving in straight direction
No
Keep boat motors moving straight
Has boat reached start end wall
No
Adjust motor to keep boat moving in straight direction
Yes
Turn off motors