The Team

Following a successful showing at the 2009 AUVSI RoboSub Competition, the St. George's Robotics Team has built a new autonomous underwater vehicle from scratch, which it is proud to enter into the 2013 competition. The robot this year is merely a skeleton of what the team hopes to accomplish in the upcoming years, however; the plan for competition this year will be more of a learning experience to see what can be improved upon in the future, in terms of both the hardware and software. This is also the first time competing in the RoboSub competition for all of the team members; regardless, despite the lack of experience, the team hopes to remain competitive in the field with this autonomous underwater vehicle, which has taken the team a total of three years to meticulously build and put together.

Spencer Chan
Electrical

Calvin Cheng
Programming

Stanley Cho
Mechanical

Evan Johnson
Programming

Ezaan Mangalji
Media

Raymond Wang
Electrical

Carl Xi
Mechanical

Leon Zhou
Mechanical

With Special Thanks:

Mr. Andrew Kay
Team Advisor

Ms. Lynette Dian
Trip Supervisor

Contributing Members:
Nicholas Lim
Spencer Louie
Giulio Rossi
Marcus Tan
Kevin Tian
Charles Wong
Charley Xi

Team Sponsors:
Whealthfields (Hong Kong) Lohmann Ltd.
Stollco Industries Ltd.
St. George's School Parents Association

The School

St. George's School is an independent high school located in Vancouver, BC. Consisting of about 700 students, the school is often known for its myriad athletic, artistic, and academic pursuits. Nonetheless, one area from which little is often heard is its robotics program. In 2009, the St. George's Robotics Team competed in the RoboSub Competition for the first time in school history, finishing 17th in the competition and earning the ``Best New Team" award. Given that the majority of the competitors were veteran teams from prestigious universities, this finish was quite an accomplishment for the St. George's team.

School Address: 4175 W 29th Avenue
Vancouver, BC
Canada V6S 1V1

School Phone: (604) 224-1304

School Email: info@stgeorges.bc.ca

School Website: www.stgeorges.bc.ca

Team Website: www.subrobotics.ca

— SINE TIMORE AUT FAVORE —
ST. GEORGE'S SCHOOL

SGS Robotics

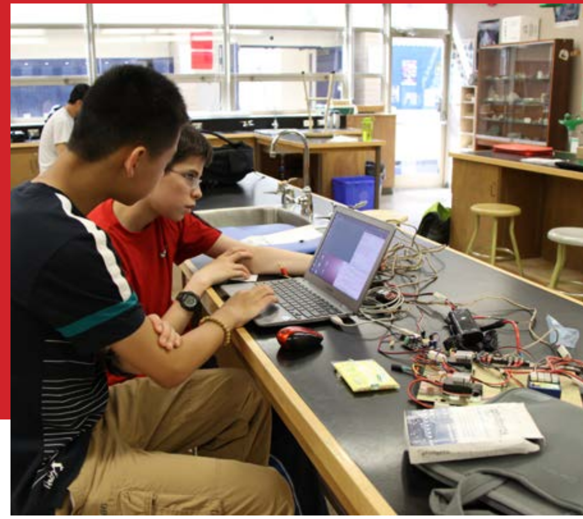
Autonomous Underwater Vehicle

AUVSI Foundation and ONR's 16th International RoboSub Competition

July 22-28, 2013

SSC Pacific TRANSDEC
San Diego, CA



Hardware

This AUV is based off the general structure of a standard dirigible airship. Though the base elements were inspired by the robot crafted by the team of 2009, a number of improvements have been added to address the shortcomings of the previous design.

The main systems are housed inside a cylindrical acrylic tube, inside which the battery compartments and main computer are stored. At one end of the tube is the compass, and at the other end, an aluminium end cap to dissipate the heat generated by the computer within the module.

Waterproof wires lead from the end cap into a thruster under an aluminium frame suspended under the main tube. This thruster turns a brass rod, which is attached to a worm gear. As the worm gear connects to the thrusters on either side of the robot, sending a signal to power the central thruster (which results in the revolution of the worm gear) will cause the vertical angle of the forward thrusters on either side to change.

This is how the depth of the robot will be controlled; the powering of the two thrusters on either side when they are tilted upwards will cause the AUV to rise, and tilting them downwards will cause the opposite to occur.

Software

Ensuring that the code ran properly was a big challenge, given the very limited testing time the team had. The programming team worked intensely to get the software working as smoothly and as quickly as possible; with only a few members having full knowledge of the API used, this was difficult, but as the team's mission strategy for the AUV this year was not overly complex, the programmers were able to make do with the little time they had.

The code for all of the software for the AUV was written in C++. In this situation, C++ was the language of choice due to its efficiency, object-oriented nature, and relative simplicity to debug and understand. Many of the programmers also come from a Java background, so the transition was fairly seamless.

All of code was compiled for the DreamPlug computer, running Debian Linux on an ARMv5 architecture. Because the DreamPlug lacked the memory and processing power to compile the code natively in an efficient manner, the team decided to cross-compile their code, storing only the binaries on the DreamPlug system. The source code was written using various text editors, and compiled via command line—because of the numerous platforms and systems that the team members used, a standard IDE was decided against.

To ensure a system, efficient approach to the code, the programming team decided to take a class-based, top-down approach to the programming. More specifically, each module that was included on the robot would have its own separate class, thus allowing a fairly simple means of organizing all of the code.

The general structure of the program was first created using a large flowchart. Then, the methods that were required to be implemented were carefully mapped out and placed onto the diagram. After the team agreed upon which classes, methods, and algorithms were to be used, each member was assigned a different portion of the code, starting from the broadest tasks (e.g. instructing the robot to begin its descent) down to the most fundamental ones (e.g. measuring and controlling the depth to which the AUV submerges).

Most of the sensor input went through the Phidget InterfaceKit 8/8/8 module, which served as a hub for both input and output. This was ideal because the InterfaceKit comprised built-in ports on API methods for reading both analogue and digital input, as well as being able to output information on both the LCD display and in the console log.

Controlling the position, orientation, and motion of the robot was the primary task, after being able to read the data input from the many sensors on the robot. The team decided to have four fundamental methods that would govern the robot: *move()*, *goDown()*, *goUp()*, and *turn()*. These four methods represent the highest level of abstraction for the AUV; the team worked down from these four methods to create a functioning robot that could complete the tasks originally set out by the mission strategy.