Motors and Motor Controller System*

Cezar Chiru
ccu12001@student.mdh.se

ABSTRACT
Every vehicle, by definition has a propulsion system which allows it to move. In the case of most AUVs (Autonomous Underwater Vehicles) the propulsion system consists of one or more thrusters. In the Naiad AUV six thrusters were used to allow the vehicle to move underwater in all three planes x, y and z, the same setup used in the Vasa project which is an earlier AUV developed by a group of students at Mälardalen University, Sweden. For the Naiad AUV a different configuration of the thrusters than the one of Vasa project was chosen to allow it to be more agile and more responsive.

1. INTRODUCTION
A requirement from the client was to make the system modular in order to be able to change the parts easily and to reduce the costs of repairs. Each of the thrusters was considered as a separate propulsion system and not as a whole. All the thrusters are inter-connected through the CAN (Controller Area Network) bus with the rest of the AUV's systems. Each of the thrusters represent a different node on the CAN bus so they are considered as separate systems. If one thruster is damaged the rest of the thrusters are not affected by that malfunction. The thrusters are given the actuation commands from their own Motor Controller.

The Generic CAN controller is a custom electronic PCB specially designed for Naiad which is composed of an AT90CAN128 micro-controller unit with a interface to the CAN bus and also to other protocols like UART, SPI, I2C which is used in some of Naiad's systems. This controller is also used to actuate the thrusters and interface them to the CAN bus.

The actuation value is computed and used to generate a PWM (Pulse Width Modulation) signal which is then transformed in a signal for the windings of the brushless DC motors used in the AUV.

*This report was written during the fall of 2013 in an advanced level project course at Mälardalen University, Sweden.

2. IMPLEMENTATION
Overview of the motor and motor controller system:

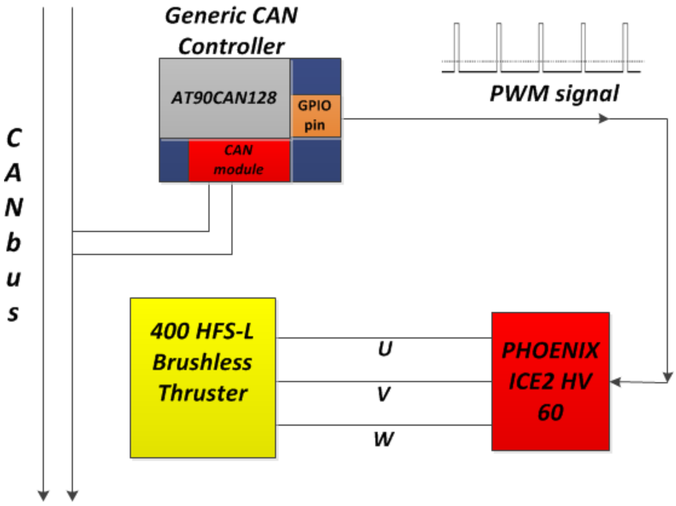

Figure 1: Motor and motor controller system overview

The implementation has the following parts:

- Thrusters
- Thruster speed controller
- CAN message processing
- Generating the PWM signal
- Main program

2.1 Thrusters
During the selection of thrusters only two models seemed to satisfy the needs of Naiad. One was the SeaBotix BTD150 which is a very popular brushed DC thruster. The other one was the CrustCrawler 400HFSL brushless DC thruster which is not as popular as the first one and a little more expensive. After the advantages and disadvantages were considered the CrustCrawler 400HFS-L has been decided to be

the better option for Naiad by being lighter, more power efficient and more powerfull than the SeaBotix BTD150.

Table 1: CrustCrwler 400 HFS-L specifications

Motor type	High efficiency brushless
Weight(motor)	185 g
Weight(thruster)	255 g
Max power	130 W
Gear ratio	4.28:1
Operating voltage	12 to 50 Volts
Depth rating	300 ft.
Thruster Housing	T-6 Aluminum
Thruster length	15.87 cm
Propeller size	60 mm

2.2 Thrusters speed controller

The CrustCrawler 400 HFS-L's brushless motor has three windings U, V and W and cannot be connected directly to the Generic CAN controller that has no feature to control brushless motors. So a special speed controller had to be used to actuate the thrusters. After a research and comparison of different brushless controllers a decision was made to go for the one suggested by the thruster manufacturer, and that is the Phoenix ICE2 HV 60 thruster controller. A feature of this brushless controller is that it has a learning function that can calibrate the controller for different systems.

The speed controllers principle of operation is to take a PWM signal as input and to convert it in signals for the brushless motor. The input of the motor controller [1] can be calibrated with the learning function by holding the throttle in the maximum position for a few seconds, then the minimum position, and finally the middle position. During the power-up the throttle of the speed controller has to be at the middle position. If during power up the controller is at the maximum forward position, the device will go into the configuration mode. By moving the throttle to different values the output will change and will cause the thruster to increase its speed or decrease it depending on what action has been taken.

2.3 CAN message processing

The speed of the thrusters is adjusted several times a second. This is done by receiving messages that contain the actuation value for the thruster from the CAN bus. The received CAN message consists of six bytes out of the maximum of 8 specified in the CAN bus.

Each of the bytes will represent an actuation value for a thruster. For example the CAN message is received by all the thrusters but each of them will use only one byte from the message. For example the thruster with id one will use the first byte, the thruster with id two will use the value from the second byte and so on. Each byte contains an eight bit signed value but the structure in which the CAN message is saved when it is received is eight bytes of unsigned values. So, for the correct interpretation of the data a conversion has to be made using the Ada Unchecked conversion feature of the Ada programming language. This feature copies the bits from one type of variable and writes them in another type of variable without raising any exceptions if the sizes of the two type are equal or the empty one is bigger.

2.4 Generating the PWM signal

The PWM signal is used to control the speed of the thrusters and also the direction of the thrust (forward or backward). The PWM signal is created by the Generic CAN controller and then fed to the brushless speed controller. The brushless speed controller expects a PWM signal with the pulse width of 1 to 2 ms, every 6 to 25 ms. This PWM signal is created by configuring some registers of the AT90CAN128 MCU.

The period of the pulse is configured with the ICR1 register and set to 16.3 ms with a resolution of 4096. To generate the pulse another register is used; OCR1A which at each Timer1 increment is compared to the TCNT1 register which keeps track of current number of clock ticks and if they match the value on the output pin is set to 0 and the TCNT1 register is set back to 0.

2.5 Main program

The main program consists of a loop in which the CAN message processing and the generation of the PWM signal is done. Inside the endless loop, the function that reads the CAN messages from the buffer is called, and if there are any available CAN messages in the buffer they will be processed. If not, it will wait for the specified amount of time for a message to be received and then skip to the next instruction.

3. RESULT

The results from the Motor Controller System have been good although not many tests have been performed on the hardware, especially because of the thrusters delivered late due to the lack of funds. The CAN communication have been tested thoroughly and the results were good. The generation of the PWM signal has been tested on the Generic CAN controller and the results were good. The brushless speed controller has also been tested and the results came out fine.

A demo has been set up and Naiad's thrusters were successfully tested although not as much as other parts of the system. One problem that might occur have and should be taken into consideration is that the thrusters were tested in air and that the first speed were they start spinning, the rotation seems to be a little faster but hopefully the friction between the water and propeller will make them spin slower so that the robot can maintain a smooth still position. Since they were not tested together underwater, adjustments and calibrations will be needed.

4. CONCLUSION

The conclusion that has been drawn is that the basic Motor Controller System is functional but more testing and calibration has to be done in order to ensure that the system will perform well underwater.

5. REFERENCES

[1] Phoenix ice hv2 controller. `http://www.castlecreations.com/products/phoenix-ice2-hv.html`. Accessed 2014-01-15.