Control Interface Design between Touch Panel and a Remotely Operated Underwater Vehicle

[1]Mehmet YAKUT, [2]Serhat YILMAZ, [3]Mehmet DÖŞLÜ
[123]Kocaeli University, Engineering Faculty, Department of Electronics and Telecommunication, Umuttepe, 41380, Kocaeli, Turkey
myakut@kocaeli.edu.tr, serhatyilmaz@ieee.org, doslumehmetcan@gmail.com

Abstract:

ROV control interfaces are generally provided by computer based programs or joystick-like devices. This study focuses on generating an independent touch panel platform for control interface according to set PWM inputs from relevant interface to ease control process of an underwater ROV. Control interface is established by graphical drawings on which sensor data are displayed and desired PWM values are set. Motion is shown upon Cartesian coordinates. Lcd.h library is used to draw desired figures and the control algorithms including communication protocols are programmed with the 'C' language. ARM Discovery Touch Screen Panel is employed to monitor the sensor data, X-Y-Z Cartesian coordinates, the sliders and the buttons. The developed interface on the panel successfully satisfies the depth and steering control requirements of the ROV.

Keywords: Control, ARM Touch Panel, Touch screen applications, ROV control interfaces

1. Introduction

Portabilities, high performances, low risks and costs of the ROVs and AUVs refer to ships and other manned vehicles, bring about an increasing interest on underwater robotics from research to industry. ROVs are capable of scientific operations in the environmental areas which have enormous cost and risks such as working in harsh topography, under ice, between corals or fields where hydrothermal events occurred [1]. However, underwater robotics is further complex with respect to ground robotics as of the control point of view, since the former should operate in unstructured environments and significant external disturbances. Nonlinear dynamic of the vehicles, narrow bandwidth of sensory and actuating systems, uncertainties in the estimation of the dynamic parameters enhance this complexity [2]. ROV control interfaces such as coastal side computers or joystick-type devices are employed for adjusting the dynamic parameters and to accomplish control tests. Design and implementation of touch panel interfaces are widely used for control applications [3, 4, 5]. Critical parameters which should be monitored and control on an underwater vehicle are its velocities and directions in the three axis and the depth [6, 7, 8].

For this purpose an experimental ROV with a test platform is developed in Kocaeli University [9]. Depth and position of the vehicle is determined by pressure sensor and electronic compass, respectively. A control card manages the depth and steering by means of PWM controlled 4 thrusters. The algorithms for the desired movements are programmed by the Touch Panel Control Interface.

This study describes the designing stages of the ROV control interface by an ARM STM32F4 DISCOVERY Touch Panel [10] which lets a platform independent from computer. This platform is easily utilized for monitoring the sensors data and adjusting the motor PWM values.

Acknowledgment : The study includes results of a project (No: 111E294) funded by the TÜBİTAK (The Scientific and Technological Research Council of Turkey). Authors would like to thank to the Council for its valuable contributions.

2. The experimental ROV designed for depth and steering control applications

The experimental platform is composed of a ROV and a water tank. The ROV's mechanics, electronic cards and software are prepared by Kocaeli University (Figure.1).

Fig.1. The ROV designed in Kocaeli University

Two thrusters are located in vertical position where the other two are located for horizontal movements.

The water tank is a plexiglass transparent material where depth and steering control applications under distorting effects (Figure.2). Disturbance is created by an 1100W submersible pump.

Fig.2. The plexiglass water tank

2.1 Control interface design between touch panel and the ROV

A hand held terminal which eliminates the requirement for carrying a computer in ROV control applications is designed. In this way, principal commands could send to the vehicle and its response could easily be monitored short of computer dependencies. The interface is prepared by STMicroelectronics Arm STM32F4 Discovery Kit and a 3.2" TFT LCD Touch Screen Panel (Fig.3). The panel reports and graphs the sensor data of the vehicle and sends PWM signals to drive the thrusters.

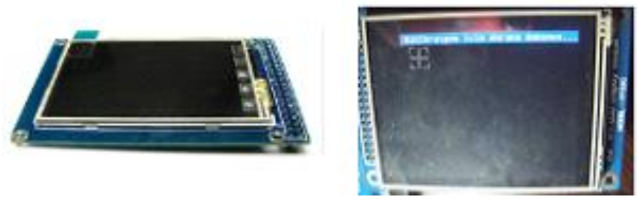

Fig.3. Touch Panel TFT LCD Module : HY32C [10]

Motions of the vehicle in each three directions are shown upon Cartesian coordinates. The panel monitors the sensor data, X-Y-Z Cartesian coordinates, the sliders and the buttons. Lcd.h library is employed to draw desired figures. The control algorithms including communication protocols and calibrations are programmed in the 'C' language. The interface also switches the lighting lambs of the ROV.

Fig.4. Prepared Graphical Interface

2.2 Calibration

While the screen has resistive features, it should be regularly calibrated by a program written in C. Three points are selected on the screen for the calculations.

(Xd, Yd) – coordinates of desired screen point
(Xs, Ys) – coordinates of suitable screen point
(A, B, C, D, E, F) – Converting factors of suitable screen point coordinates to relevant points.
The points touched on the screen is calculated by Eq.1. and displayed on the screen.

$$\begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \begin{bmatrix} A & B & C \\ D & E & F \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ 1 \end{bmatrix} \qquad (1)$$

2.3 PWM Control by Sliders

Sliders graphed on the display adjust PWM rates which drive the thruster motors whose directions are selectable. M1, M2, M3, M4 sliders indicate the Vertical Left, Vertical Right, Horizontal Left and Horizontal Right Motors, respectively (Fig.4). Sliders are drawn due to the entered coordinates mentioned in Eq.1 by LCD_Drawline() function which refers Bresenham line algorithm [11]. PWM values are increased and decreased by grades of ten percent from 0% to 100%.

Fig.5. Adjusting Motor Speeds by Sliders

2.4 Monitoring the Sensors Data

Sensors data could be plotted in X-Y-Z coordinates. Z indicates the depth value. Sensors data are acquired with I2C serial communication protocol. Configuration and monitoring codes are given in Appendix.A

3. Results and discussion

The Control Interface has driven the ROV by PWM signals which use 42 Khz frequency and in %70 to %20 full/empty rates. PWM outputs are checked by an Oscilloscope (Fig.6).

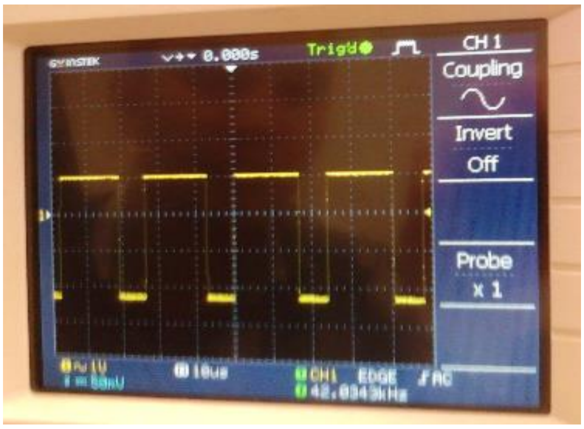

Fig.6. %70 PWM Rate from PA0 port which drives the M1.

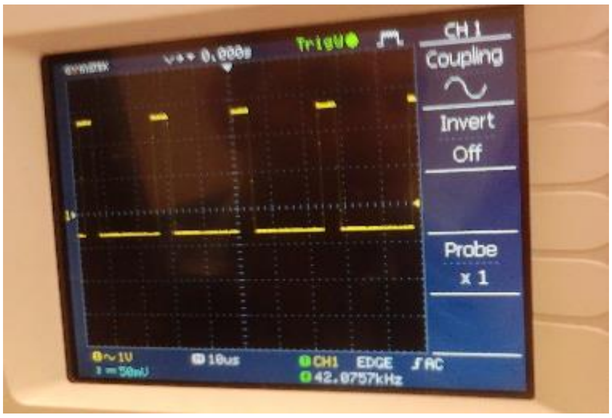

Fig.7. %70 PWM Rate from PA1 port which drives the M2.

The ROV is tested by several motor speed combinations driven by the PWM signals from the designed terminal (Fig.8, Fig.9). When M1 and M2 are effective, the ROV moves in vertical direction, vice versa.

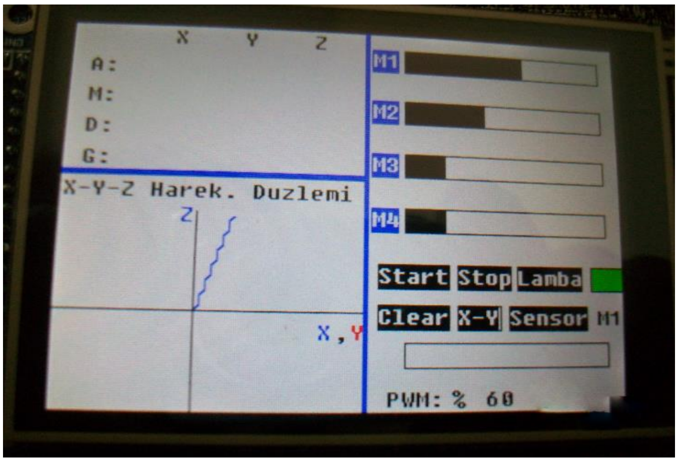

Fig.8. Vertical motors are effective

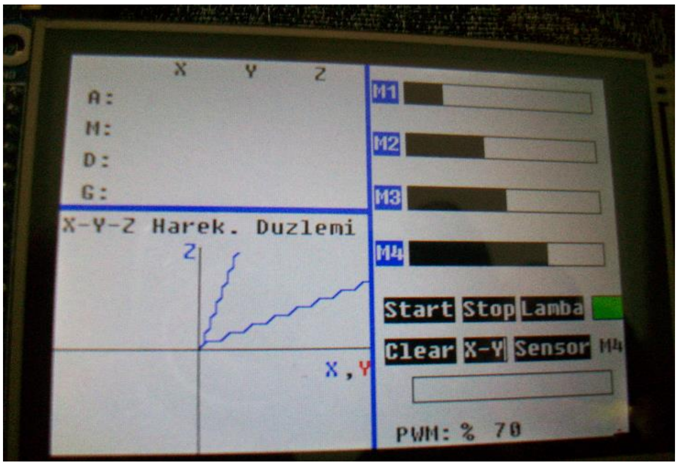

Fig.9. Horizontal motors are effective

In any occasions, The ROV responds to the terminals commands (Fig.10,Fig.11)

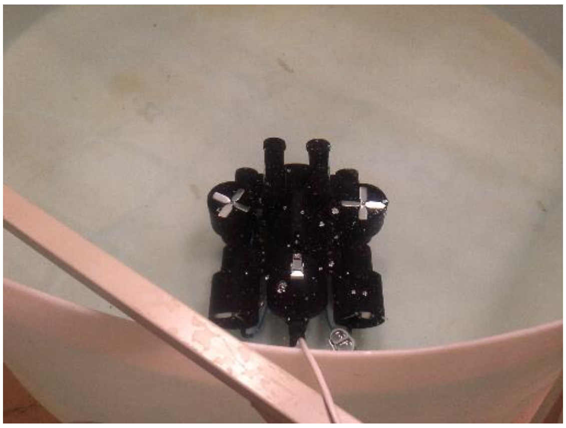

Fig.10. Horizontal control tests by the terminal

Fig.11. Vertical and mixed tests of the ROV

4. Conclusion

Designing stages of a control interface terminal which eliminates the requirement for carrying a computer in ROV control applications is given. The ROV and its test platform is designed for depth and steering control applications and researching of novel control algorithms and devices. Programs for calibration, communication, monitoring and control for the touch panel interface is prepared and tested. For detecting the touched point precisely, a more fault tolerant formula could be developed. Capacitive touch panels may be examined in the future studies. ROV camera outputs can be monitored for heavier missions. Nevertheless, the designed interface on the panel satisfies the depth and steering control needs of the system.

REFERENCES

[1] Meinecke G, Ratmeyer V, Renken J. HYBRID-ROV, Development of a new underwater vehicle for high-risk areas. Oceans 2011 IEEE, pp. 1-6, September 2011.
[2] Antonelli G. Underwater Robots-Motion and Force Control of Vehicle-Manipulator Systems. 2nd ed., vol.2. Springer: Berlin Heidelberg New York, 2006.
[3] Bai YW and Lin HE. Design and Implementation a Control Interface by Touch Panel for an Embedded Arbitrary Waveform Generator, SICE Annual Conference 2008, pp. 2811-2816, August 2008.
[4] Bai YW, Lin HE, Jau WC. Design and Implementation of an Integrated Man-Machine Interface by Touch Panel for an Embedded Electronic Measurement System. VECIMS 2008 – IEEE International Conference onVirtual Environments, Human-Computer Interfaces, and Measurement Systems, pp.31-36, July 2008
[5] Sekimoto T, Tsubouchi T, Yuta S, An Implementation of a Human Interface using a Touch Panel and Visual Images for Controlling a Vehicle. RO-MAN'96, Robot and Human Communication, 5th IEEE International Workshop on, pp. 451-455,Tsukuba, November 1996.
[6] Inoue T, Shibuya K, Nagano A. Underwater Robot with a Buoyancy Control System Based on the Spermaceti Oil Hypothesis - Development of the Depth Control System. The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18-22, October 2010.
[7] Kim J, Kim K, Choi HS, Seong W, Lee KY. Depth and Heading Control for Autonomous Underwater Vehicle Using Estimated Hydrodynamic Coefficients. MTS/IEEE OCEANS 2001 Conference and Exhibition, Honolulu Hawaii, vol.1, 429-435, November 2001.
[8] Kim HS,Shin YK. Design of Adaptive Fuzzy Sliding Mode Controller using FBFE for UFV Depth Control. The 2006 SICE-ICASE International Joint Conference, Bexco-Busan, Korea, 3100 – 3103, 18-21 October 2006.
[9] Yilmaz S, Yakut M, Ince S. Experimental ROV Design for Depth and Steering Control Applications. The Scientific and Technological Research Council of Turkey (TÜBİTAK), 1002 Short-Term R&D Project, Final Report, Project No:111E294, July 2013, Kocaeli/Turkey
[10] http://www.st.com/web/en/catalog/tools/FM116/SC959/SS1532/PF250863,Visiting Date: 22 August 2013
[11] http://en.wikipedia.org/wiki/Bresenham's_line_algorithm, Visiting Date: 21 August 2013

Appendix

```
void SensorKonfig()
{
  I2CInit();
 ADXLInit();
```

```
 gyroEnableDefault();
 HMCInit();
}

void SensorData()
{
   ADXLRead(AcceE);  //Reading 3 axis Accelerometer data
                     sprintf(geciciler, "", AcceE[0]);   //X Axis
                    GUI_Text(12, 20, (uint8_t*)geciciler, Black, White);
                     sprintf(geciciler, "", AcceE[1]);   //Y Axis
                    GUI_Text(22, 20, (uint8_t*)geciciler, Black, White);
                     sprintf(geciciler, "", AcceE[2]); //Z Axis
                    GUI_Text(32, 20, (uint8_t*)geciciler, Black, White);
                    gyroRead(GyroE);     //Monitoring Gyro Data
                     sprintf(geciciler, "", GyroE[0]);  //X Axis
                    GUI_Text(12, 40, (uint8_t*)geciciler, Black, White);
                     sprintf(geciciler, "", GyroE[1]);  //Y Axis
                    GUI_Text(22, 40, (uint8_t*)geciciler, Black, White);
                     sprintf(geciciler, "", GyroE[2]);  //Z Axis
                    GUI_Text(32, 40, (uint8_t*)geciciler, Black, White);
                    HMCRead(MagneE);  //Reading Magnetometer
                     sprintf(geciciler, "", MagneE[0]); //X Eksen
                    GUI_Text(12, 80, (uint8_t*)geciciler, Black, White);
                     sprintf(geciciler, "", MagneE[1]);  //Y Eksen
                    GUI_Text(22, 80, (uint8_t*)geciciler, Black, White);
                     sprintf(geciciler, "", MagneE[2]); //Z Eksen
                    GUI_Text(32, 80, (uint8_t*)geciciler, Black, White);
}
```

Defined Variables are

```
int16_t AcceE[3];
int16_t GyroE[3];
int16_t MagneE[3];
 int8_t  geciciler[20]={0};
```