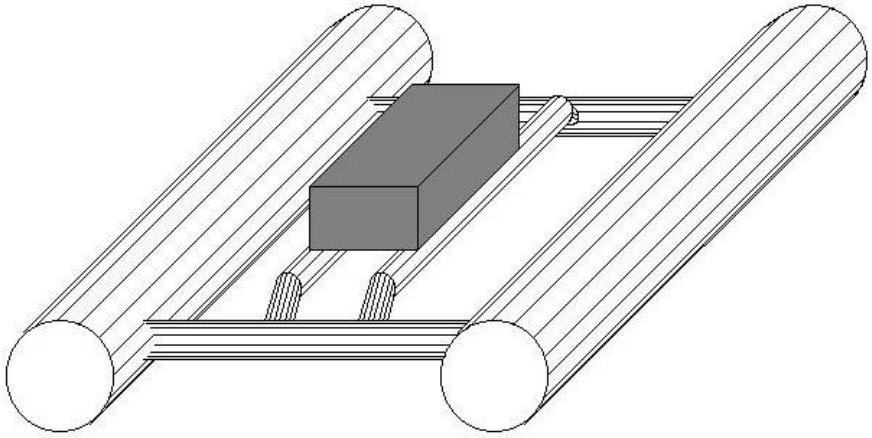
UNIVERSITY OF FLORIDA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
EEL5666
INTELLIGENT MACHINE DESIGN LABORATORY



APB – THE *A*UTONOMOUS *P*ONTOON *B*OAT

Daniel Williams
Instructors: Dr. A. Antonio Arroyo and Dr. Eric M. Schwartz
TA's: Scott Jantz and Patrick O'Malley
August 4th, 1999

TABLE OF CONTENTS

Abstract 3

Executive Summary 4

Introduction 5

Integrated System 6

Mobile Platform 7

Actuation 8

Sensors 9

Behaviors 13

Conclusion 14

Vendors 15

Appendix A 16

Appendix B 19

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

ABSTRACT

APB is based on the structure of a pontoon boat. The structure will consist of two parallel tubes connected by perpendicular tubes in the front and back. Bow-thrust motors will be used for forward, reverse and rotational propulsion.

Sonar will be used for collision avoidance. A microphone will be used for bump detection. A water sensor will be placed inside each of the boat's electronic compartments to detect for water leaks and the boat will exhibit a warning behavior if water is detected.

APB will use underwater sonar to track the deepest point in the body of water. The robot will report the current maximum depth discovered through a dual seven-segment LED display.

Behavioral programming will be done using the Interactive C compiler.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

EXECUTIVE SUMMARY

APB's primary function is to patrol a body of water, taking depth measurements. System control is maintained by a single Motorola MC68HC11 micro-controller, mounted on the MC68HC11EVBU board, with the Mekatronix ME11 expansion board. The structure of the boat consists of 3" PVC acting as the pontoons, connected by ½" PVC.

In order to allow the boat to carry all the required electronics and power supplies, a frame of substantial size had to be developed. Due of its size of approximately 34" in length and 21" in width, the weight reduces the motors' effective thrust-to-weight ratio, but the boat is still able to maintain an adequate "slow-walking" speed.

Building a surface water vehicle provides interesting challenges. First, navigating is not as precise as on land because of drift currents. In water, the mechanics of stopping and turning are less precise because the boat is counteracting the momentum of the water trailing the boat, which is continually pushing the boat in the direction of the previous heading.

Water sensors built into each compartment help provide early warnings of water leakage so that the boat can be retrieved before permanent damage occurs.

Finally, the Standard Communications DS100 depth sensor is used to record the water depth as the boat patrols around. This role as a 'science vessel' is the main purpose of the boat.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

INTRODUCTION

APB is designed to be an autonomous pontoon boat that randomly patrols a body of water in search of the greatest depth. This maximum depth will be recorded internally in memory of the Motorola MC68HC11E9FN, the main CPU of the robot. Upon user interaction (switching on the dual seven-segment LED output), the display on the robot will display the currently known maximum depth.

A primary concern in building APB is ensuring that the mechanical structure *floats*. Building a water based robot removes the freedom to load any and all equipment on the structure (at the sacrifice of speed) as in land-based robots.

The following document describes each component system of APB. A high level description of the robot will be given followed by a breakdown into smaller sections consisting of the mechanical structure, actuation/propulsion, sensors, behaviors and conclusion.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

INTEGRATED SYSTEM

APB is controlled by the Motorola MC68HC11E9 processor. The system consists of six input circuits: two collision sensors (microphones), a water sensor, sonar ranging module and sonar depth module. The outputs consist of two motor driver circuits and a dual seven-segment LED display for depth output.

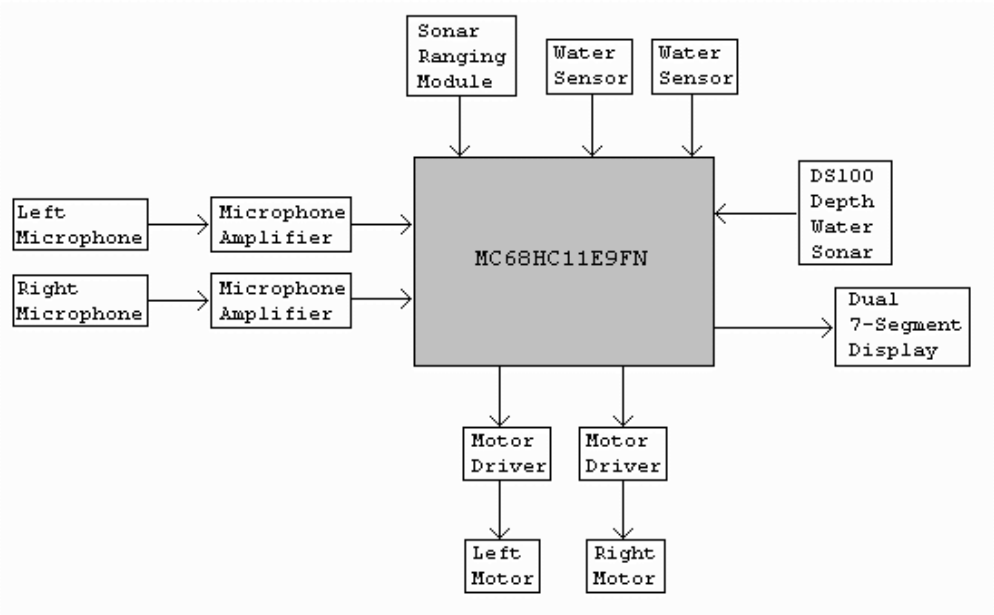


FIGURE 1: Block diagram of integrated system.

The two independent motor driver circuits are comprised of a relay for direction control and a power MOSFET for pulse width modulation (PWM). Each motor driver is capable of delivering a maximum of 5Amps of constant current.

The microphone circuit consists of a basic audio amplifier (LM386) and an analog comparator. The comparator output then drives a D-FF that is connected to a memory-mapped I/O address. The microprocessor can then read that I/O address to determine if a collision has occurred.

The water sensor simply consists of two metal contacts in close proximity (but not touching). If water is present on the sensor, the two contacts show a measurable resistance, and thus are used to drive an inverter buffer, for which the output connects to an input port to the 68HC11.

The sonar module is straightforward – it is calibrated for a maximum distance of approximately ten feet and then increments of that distance are registered as a voltage from 0V to 5V.

The depth sensor, made and donated by Standard Communications, has a serial output compatible with the NMEA-0183 standard. The NMEA-0183 is a GPS/marine product serial format standard similar to the PC RS-232 standard.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

MOBILE PLATFORM

The robot platform consists of a PVC frame. The motors are mounted directly to this frame, in the middle of each pontoon. Figure 2 shows the basic view of the platform.

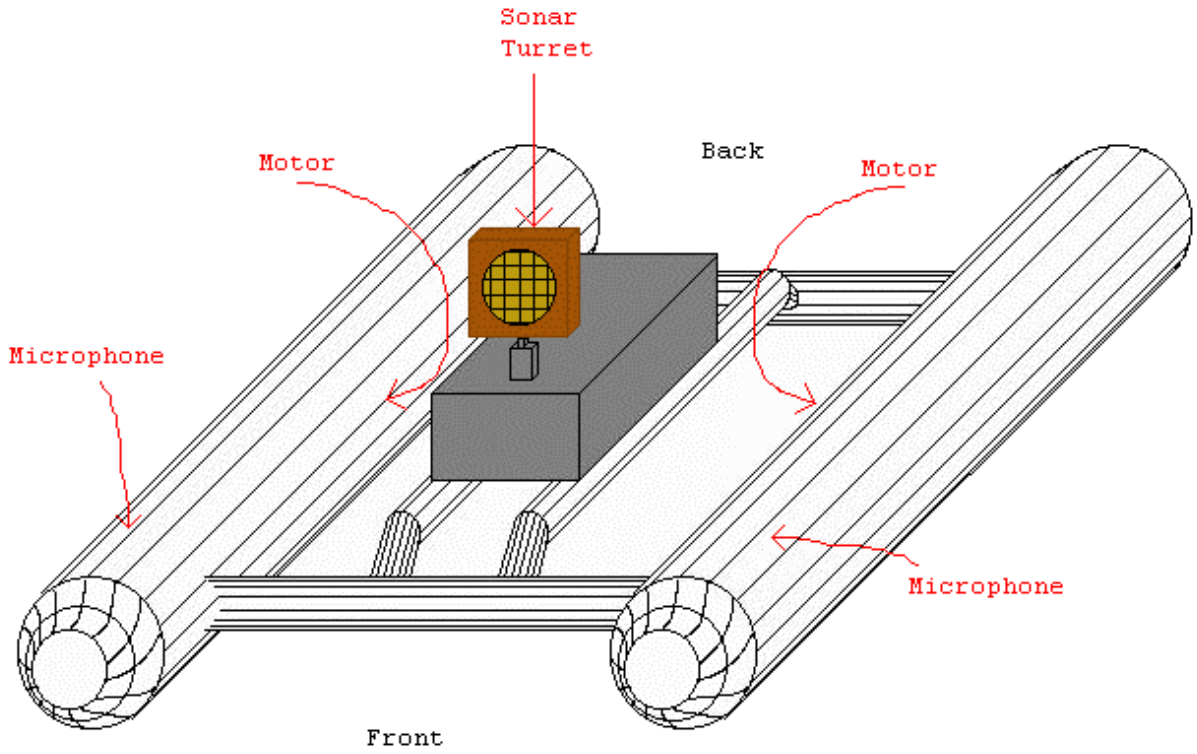


FIGURE 2: Pontoon Boat basic structure

The pontoons are made of 3" PVC pipe, and are each approximately 30" long. The pontoons are interconnected by ½" PVC pipe, approximately 1' in length. A support structure for the electronics is also constructed of ½" PVC pipe, raised slightly from the cross beams to avoid drag in the water.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

ACTUATION

BATTERY POWER

The microprocessor and sensor circuits are powered by an 8pk of 1000mAh hi-capacity AA NiCAD batteries. The motors and sonar turret servo are powered by two 7.2V 1500mAh NiCAD battery packs.

MOTORS

The propulsion of the robot is made possible by two Graupner Speed 400 bow thruster motors. Each motor draws a maximum of 3.5A of constant current. A picture of the motor is shown in Figure 3.

FIGURE 3: The Graupner Speed 400 bow thruster motor.

Each motor driver consists of one MTP60N06 power MOSFET for PWM speed control and one 5V DPDT relay (contacts rated at 5A) for direction control. The MOSFET PWM signal is driven by an output compare port and the direction is from any normal output port. For the PWM signal, a 0V signal will turn off the motor and a 5V signal will run the motor at full speed. Increments in the duty cycle will proportionally affect motor speed. The direction signal uses 0V for forward motion, and 5V for reverse motion. All signals are opto-isolated to protect the micro-controller. A schematic of the motor driver is shown below in Figure 4.

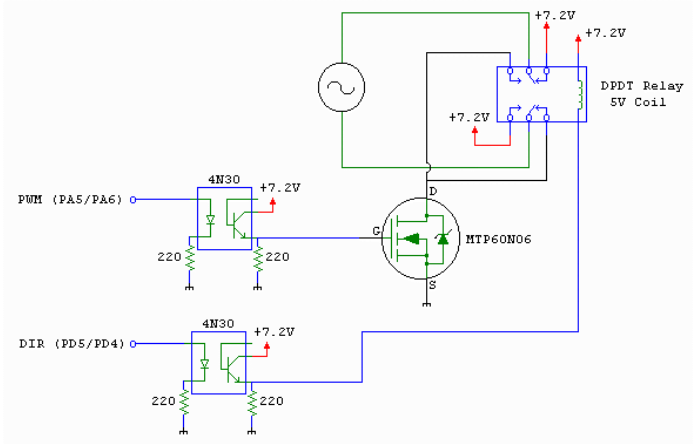


FIGURE 4: Motor driver used to control motor speed and direction.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

SENSORS

There are approximately six sensors on the APB robot. A table of sensors with their functions are listed below:

Quantity	Sensor	Function
2	microphone	collision detection
2	water sensor	water detection
1	sonar ranging unit	collision avoidance
1	sonar depth ranging unit	calculates water depth

COLLISION DETECTION:

The circuit in Figure 5 is used to detect sudden hull vibrations using a microphone, analog comparator, logic inverter and D-flip-flop.

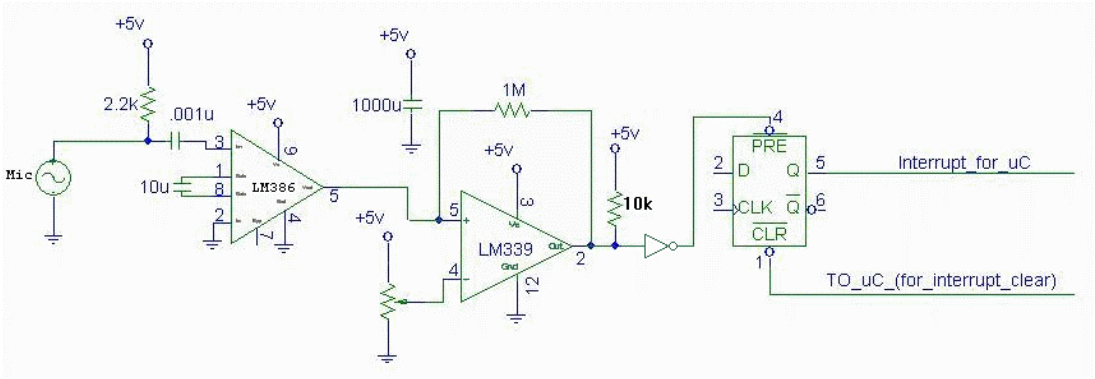


Figure 5: Collision detection circuit (x2, left and right).

The LM386 audio amplifier amplifies the signal from the microphone and then the LM339 analog comparator compares it with a set threshold voltage (adjustable by the pot). If the microphone peaks at a voltage greater than the threshold (set by the potentiometer), the D-FF gets set. The 68HC11 then reads this signal from a memory-mapped I/O port (to process a collision behavior) and then issues a low-True signal to the clear of the D-FF to reset it.

COLLISION AVOIDANCE:

For collision avoidance, I am using the SonaSwitch Mini-S sonar unit. This sonar unit is mounted atop a servo for left-right rotation. The sonar unit has been calibrated for a maximum distance of approximately 11'. Thus, the sonar registers approximately 0.45V for every foot of distance up to the maximum. Figure 6 shows the data for the sonar, as measured with a multi-meter.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

Range	Voltage
1'	0.60V
2'	1.02V
3'	1.46V
4'	1.80V
5'	2.30V
6'	2.70V
7'	3.10V
8'	3.60V
9'	4.00V
10'	4.50V
11'	5.10V

Figure 6: Sonar module data measurements.

The sonar module produces a relatively consistent linear output.

WATER SENSOR:

The water sensor is simple – two metal contacts mounted in parallel on the bottom of the circuitry box. This way, if any water leaks into the box, the two contacts produce a resistance which will be used to drive a logic inverter which drives a bit on an input port to the 68HC11. The circuit used for the water sensor is shown below in Figure 7.

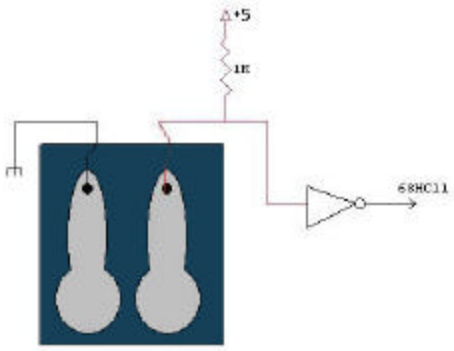


Figure 7: Water sensor.

Depending on the ion concentration of the water, the contacts can generate resistances of 20kΩ and up.

SONAR DEPTH RANGING UNIT:

Manufactured and donated by Standard Communications, the DS100 Depth Sounder is an NMEA-0183 output compatible device capable of sensing depth from 3' to 400'. It has built in damping factors to help filter out inaccurate readings resulting from bouncing waves off of fish

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

and other randomly moving objects. It also has turbulence filters to help eliminate inaccurate readings resulting from motor noise and air bubbles. Finally, it requires a 200kHz transducer which must be submerged. The transducer acquired for this project was the Standard Communications DST51 Transom Mount transducer. The DS100 is shown in Figure 8.

FIGURE 8: The Standard Communications DS100 Depth Sounder

The serial output of the DS100 exists in the range of normal TTL-level signals, i.e. 0-5V. For this reason, it cannot be directly connected to the serial connector of the MC68HC11EVBU board. This is because the serial connector is fed into the Motorola MC145407 RS-232 interface chip, which requires standard RS-232 level voltages, i.e. in the range of –12V to –3V and +3 to +12V. Instead, the serial output of the DS100 is fed through a logic inverter (the serial signal is produced in inverted form, like the RS-232 standard), and then fed directly into the MC68HC11 serial-in pin (Port D pin 0 RXD), which accepts 0-5V signals.

In order to avoid data collision between the output of the DS100 and the output of the MC145407 chip, a trace on the MC68HC11EVBU board, jumper J8, had to be cut and a switch placed over the jumper J8. This allowed the serial inputs to be switched one way for programming the 68HC11, and the other way for input from the DS100.

BRIEF SUMMARY OF THE NMEA-0183 FORMAT

The NMEA-0183 standard outputs data in the form of ASCII 'sentences'. For depth sounders such as the DS100, a sample sentence is shown below:

```
   .
   .
$SDDPT,,
$PTTKD,0023,A,C
$SDDBT,023.9,f,,,,
$SDDPT,,
   .
   .
```

The NMEA-0183 sentence format is composed of four main parts: (1) The dollar sign '$' to denote the beginning of a new sentence, (2) the talker identification keyword (i.e., the SD means

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

Sounder, Depth), (3) the sentence identification keyword (i.e., the DBT means Depth Below Transducer), and finally (4) the data fields, which are delimited by commas.

For the purposes of this project, the sentence of interest is the third sentence from the above sample, namely the sentence beginning with $SDDBT. This sentence reports the depth sounder's depth below the transducer, which is actual depth of the water. The data field shows that the transducer waves calculate to a depth of 023.9, followed by 'f' for feet. Notice the four commas after the 'f'. These fields are empty because the depth sounder is set to report the depth in feet. If it were set to display in meters, the depth sounder would report a sentence in the form of:

$SDDBT,,,007.2,m,,

and if it were set to fathoms, it would report a sentence in the form of:

$SDDBT,,,,,3.9,F

The data is captured into the 68HC11 simply by performing reads of the serial input buffer after a complete byte has been received.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

BEHAVIORS

The APB robot's main behavior is to randomly patrol a body of water in search of the maximum depth. It accomplishes this by using sonar for obstacle avoidance, and microphones for collision detection. The water sensors, if activated, initiate a behavior which stops the boat and sends an alert by flashing LEDs. The depth is monitored by the Standard Communications DS100 depth sounder attached to their DST51 Transom Mount depth transducer.

Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

CONCLUSIONS

APB ultimately proved to be a reliable autonomous water-roaming robot. The sonar unit used for obstacle avoidance worked remarkably well, as did the microphones (after many adjustments).

The most time consuming part of the project was fixing the recurring problems with the motor drivers, namely problems with the power transistors not working correctly and/or overheating. Unfortunately, due to lack of time from fixing the recurring motor driver problems, successful retrieval of the depth information from the DS100 depth sounder was never successfully accomplished.

The potential uses for a boat of this type are too numerous to count. With more powerful motors, longer battery life, more precise navigation and a more waterproof structure, this boat could be used to do floor mapping and other forms of data acquisition and analysis.

Future enhancements I would like to make to APB include adding more behaviors, such as one that would make it return to the shore after the battery life began to drop below a desired level. I would also like to add a speaker circuit to convey audible 'pings' so one can hear the relative depth of the water from a distance. More sensors could be added, such as one to monitor water temperature. Finally, I would like to add RF remote control capability so the robot's current behavior and/or navigation headings can be reassigned without retrieving it and reprogramming.

VENDORS

COMPANY	PHONE NUMBER	PARTS ORDERED
Standard Communications	1-800-767-2450	DS100 Depth Sounder
Digikey	1-800-344-4539	5V DPDT 5A Relay
Hobby Lobby	1-615-373-1444	Graupner Speed 400 Bow Thruster Motors
Marlin P. Jones and Assoc.	1-800-652-6733	Various sensor kits
Mekatronix		ME11 Expansion Board, servo, sonar unit

APPENDIX A

```
/*******************************************************
**                                                    **
** Title:        APB_Rev1.c                           **
** Programmer:   Daniel Williams                      **
** Date:         July 29th, 1999                      **
**                                                    **
*******************************************************
**                                                    **
** APB is an autonomous pontoon boat that wanders     **
** around a body of water and takes depth measurements **
** using the Standard Communications DS100 depth      **
** sounder (with NMEA interface).                     **
**                                                    **
** APB uses sonar for obstacle avoidance and          **
** microphones for collision detection. Water sensors **
** are placed at the bottom of each compartment to    **
** detect leaks.                                      **
**                                                    **
** Two seven-segment LED displays are mounted on the  **
** exterior of the boat to report a maximum depth of  **
** 0-99 feet.                                         **
**                                                    **
**                                                    **
** Future Robot Improvements:                         **
**                                                    **
**  (1) Audible "pings" to convey water depth from a  **
**      distance.                                     **
**                                                    **
**  (2) Some form of remote control to tell the boat to **
**      return to shore. MPJA has a remote control    **
**      relay kit (UHF) for $34.                      **
**                                                    **
*******************************************************/


/******** Defines ***************************/
#define SENSORS    peek(0x4000) & 0x0f
#define SONAR      analog(0)
#define ERROR      10

#define LED_LMOTORF    0x80
#define LED_LMOTORB    0x40
#define LED_RMOTORF    0x20
#define LED_RMOTORB    0x10

/***** End of Defines ***********************/

int   LEFT_MOTOR   = 0     ;
int   RIGHT_MOTOR  = 1     ;
float MAX_SPEED    = 100.0 ;
float ZERO_SPEED   = 0.0   ;
float SERVO_LEFT   = 45.0  ;
float SERVO_CENTER = 85.0  ;
float SERVO_RIGHT  = 135.0 ;

/******* NAVIGATION COMMANDS *******/
void ALL_AHEAD  ( )
  { motor( LEFT_MOTOR,  100.0 ); motor( RIGHT_MOTOR,  100.0 );
    PRINT_LED_DISPLAY( LED_LMOTORF | LED_RMOTORF ); }

void ALL_BACK ( )
  { motor( LEFT_MOTOR, -100.0 ); motor( RIGHT_MOTOR, -100.0 );
    PRINT_LED_DISPLAY( LED_LMOTORB | LED_RMOTORB ); }
```


Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

```
void ALL_STOP ( )
  { motor( LEFT_MOTOR,   0.0 ); motor( RIGHT_MOTOR,   0.0 );
    PRINT_LED_DISPLAY( LED_LMOTORF | LED_LMOTORB | LED_RMOTORF | LED_RMOTORB ); }

void TURN_RIGHT ( )
  { motor( LEFT_MOTOR,  100.0 ); motor( RIGHT_MOTOR, -100.0 );
    PRINT_LED_DISPLAY( LED_LMOTORF | LED_RMOTORB ); }

void TURN_LEFT ( )
  { motor( LEFT_MOTOR, -100.0 ); motor( RIGHT_MOTOR,  100.0 );
    PRINT_LED_DISPLAY( LED_LMOTORB | LED_RMOTORF ); }
/***** END NAVIGATION COMMANDS *****/

/****** SONAR TURRET CONTROLS ******/
void LOOK_LEFT   () { servo_deg2( SERVO_LEFT   ); }
void LOOK_CENTER () { servo_deg2( SERVO_CENTER ); }
void LOOK_RIGHT  () { servo_deg2( SERVO_RIGHT  ); }
/**** END SONAR TURRET CONTROLS ****/


void CLEAR_MIC()     /* Resets FFs and the deasserts Reset. */
  { poke(0x4000,0x00); poke(0x4000,0xff); }

void PRINT_LED_DISPLAY( int BIT )    /* Useful for printing updated motor and sensor status. */
  { poke( 0x5000, 0x00 | BIT | SENSORS ) ;}

void main()
{
  int center, left, right, mic;

  init_motors();

  CLEAR_MIC();                 /* Since the Q for D-FFs are on at powerup, clear them. */
  PRINT_LED_DISPLAY( 0 );      /* Initialize LED display to initial state. */
  servo_on();                  /* Turn on sonar turret. */
  LOOK_CENTER();  sleep(3.0);  /* Re-center turret. */
  ALL_AHEAD();    sleep(1.0);  /* For test purposes to let motors get rolling. */

  while(1)
    {

      while( !(SENSORS & 0x03) )   /* While microphones register zero... */
        {
          if( SENSORS & 0x0c )   /* Water Sensors */
            {
              ALL_STOP();
              while(1)
                {
                  poke(0x5000,0xff); sleep(0.25);
                  poke(0x5000,0x00); sleep(0.25);
                }
            }

          /* Obstacle avoidance */
          if( SONAR < 100 )
            {
              ALL_STOP();
              sleep(1.0);

              LOOK_LEFT();    sleep(1.0);  left   = SONAR;  sleep(1.0);
              LOOK_RIGHT();   sleep(1.0);  right  = SONAR;  sleep(1.0);
              LOOK_CENTER();  sleep(1.0);  center = SONAR;  sleep(1.0);

              if ( left > (right+ERROR) )     /* Right side is closer, curve left. */
                 {  TURN_LEFT();  sleep(5.0); ALL_STOP();  sleep(1.0); ALL_AHEAD(); }
              else if( right > (left+ERROR) ) /* Left side is closer, curve right. */
                 {  TURN_RIGHT(); sleep(5.0); ALL_STOP();  sleep(1.0); ALL_AHEAD(); }
```


Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

```
            else  /* In questionable zone, back up and turn around. */
               {  ALL_BACK();   sleep(5.0); ALL_STOP();  sleep(1.0);
                  TURN_RIGHT(); sleep(8.0); ALL_STOP();  sleep(1.0); ALL_AHEAD(); }

            CLEAR_MIC(); PRINT_LED_DISPLAY( LED_LMOTORF | LED_RMOTORF );
           }
        }

      mic = (SENSORS & 0x03);
      CLEAR_MIC();

      ALL_STOP(); sleep(1.0);
      ALL_BACK(); sleep(5.0);
      ALL_STOP(); sleep(1.0);

      if( mic & 0x01 )        /* Left Mic detected bump */
        { TURN_RIGHT(); sleep(6.0); ALL_STOP(); sleep(1.0); ALL_AHEAD(); }
      else if( mic & 0x02 )   /* Right Mic detected bump */
        { TURN_LEFT();  sleep(6.0); ALL_STOP(); sleep(1.0); ALL_AHEAD(); }
      else  /* Both or No Mics detected bump just turn around */
        { TURN_RIGHT(); sleep(8.0); ALL_STOP(); sleep(1.0); ALL_AHEAD(); }

      poke(0x5000,0xff); sleep(0.25);
      poke(0x5000,0x00); sleep(0.25);

      ALL_AHEAD();
      sleep(0.5);
      CLEAR_MIC();
    }

print("SHOULD NEVER EVER BE HERE %d\n", SENSORS);

}
```


Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley

APPENDIX B

Though the depth sounder was never successfully interfaced, this is the code I wrote to retrieve the depth data.

```
/*                                        */
/* Daniel Williams                        */
/*                                        */
/* NMEA Capture Routines (DS set to feet)  */
/*                                        */
/* NMEA sentence structure produced by the */
/* depth sounder is as follows:           */
/*       .                                */
/*       .                                */
/*   $SDDBT,012.1,f,,,,                   */
/*   $SDDPT,,                             */
/*   $PTTKD,16.0,A,C                      */
/*   $SDDBT,012.1,f,,,,                   */
/*       .                                */
/*       .                                */

#define _DOLLAR 0x24
#define _COMMA  0x2C
#define _B      0x42
#define _D      0x44
#define _S      0x53
#define _T      0x54

int NMEADepth()
  {
    int i = 0, x = 0;
    int talkID[2];
    int sentID[3];
    int depth[5];

    while(1)
      {
        x = get_char();
        if( x == _DOLLAR )
          {
            talkID[0] = get_char();
            talkID[1] = get_char();
            if( (talkID[0] == _S) && (talkID[1] == _D) )
              {
                sentID[0] = get_char();
                sentID[1] = get_char();
                sentID[2] = get_char();
                if( (sentID[0] == _D) && (sentID[1] == _B) && (sentID[2] == _T) )
                  {
                    x = get_char();  /* Grab comma */
                    depth[0] = get_char();
                    depth[1] = get_char();
                    depth[2] = get_char();
                    depth[3] = get_char();
                    depth[4] = get_char();

                    for( i=0; i < 5; i++) { depth[i] -= 48; }  /* Convert ASCII numbers to HEX */

                    return (depth[2] + 10*depth[1]);   /* Only want depths from 0-99 feet. */
                  }
              }
          }
      }
  }
```


Daniel Williams
Aug 4th, 1999
Instructors: Dr. Arroyo
Dr. Schwartz
TA's: Scott Jantz
Patrick O'Malley