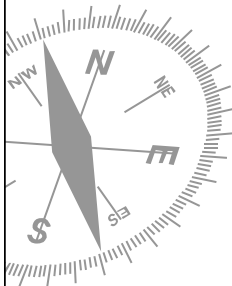




# Obstacle Avoidance using the EGN-1935 Robots

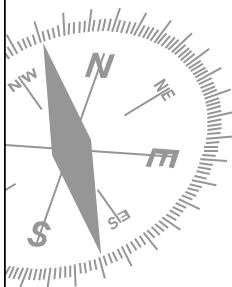
A. A. Arroyo



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo



# IMDL Software Series Assembler



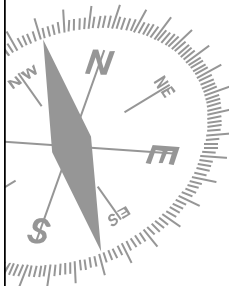
University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

test_EGN_1935_robot.asm
Title: Test Atmel128 Board on EGN-1935 Robots
Programmer: A. Antonio Arroyo
Date: September 24, 2013
Version: 1.0
Description:
  * Test the Atmel ATMEGA128 board on EGN-1935 robots w/ the Assembler
  * Connect two LEDs to Pins PAB & PAI with limiting resistors
  * LEDs alternate (on,off), (off,on), ... (on,off) = the value in R16
  * =====
  * Include "m128def.inc" include an Atmega128 compatible definitions file
  * Include "m128def.inc" The real Atmega128 definitions file
  * =====
  * Place the following code from address 0x0000
  * =====
  * Jump RESET
  * =====
RESET:
  ;Reset Label (start of Main)
  ;Store 10 in R16 (Repeat Count)
  ldi R16, 0x0A
  ;Store 50 in R17 (i, Outer Loop)
  ldi R17, 0x32
  ;Store 255 in R18 (j, Middle Loop)
  ldi R18, 0xFF
  ;Store 255 in R19 (k, Inner Loop)
  ldi R19, 0xFF
  ;Set PAB and PAI to output (HIGH)
  ldi R20, 0x03
  out DDRA, R20
  ;Set PORTA pins PAB-PAI high
  ldi R20, 0x01
  out PORTA, R20
  ;Write Data to PORTA
  ;Delay Loop 1
  ;Decrement R19
  dec R19
  ;If not zero jump to the Loop Label
  brne Loop1
  ;Restore 255 in R19
  ldi R19, 0xFF
  ;Decrement R18
  dec R18
  ;If not zero jump to the Loop Label
  brne Loop1
  ;Restore 255 in R18
  ldi R18, 0xFF
  ;Decrement R17
  dec R17
  ;If not zero jump to the Loop Label
  brne Loop1
  ;Restore 255 in R17
  ldi R17, 0x32
  ;Store 0xFF in R18
  ldi R18, 0xFF
  ;Store 0xFF in R19
  ldi R19, 0xFF
  ;Set PAB Low and PAI High
  ldi R20, 0x02
  out PORTA, R20
  ;Write Data to PORTA
  ;Delay Loop 2
  ;Decrement R19
  dec R19
  ;If not zero jump to the Loop Label
  brne Loop2
  ;Restore 255 in R19
  ldi R19, 0xFF
  ;Decrement R18
  dec R18
  ;If not zero jump to the Loop Label
  brne Loop2
  ;Restore 255 in R18
  ldi R18, 0xFF
  ;Decrement R17
  dec R17
  ;If not zero jump to the Loop Label
  brne Loop2
  ;Restore 255 in R17
  ldi R17, 0x32
  ;Restore 50 to R17
  ldi R17, 0x32
  ;Restore 255 to R18
  ldi R18, 0xFF
  ;Restore 255 to R19
  ldi R19, 0xFF
  ;Set PAB High, PAI Low
  ldi R20, 0x01
  out PORTA, R20
  ;Write Data to PORTA
  ;Decrement COUNT
  dec R16
  ;If not zero jump to the Loop Label
  brne Loop1
  ;Set PAB and PAI HIGH to finish
  ldi R20, 0x03
  out PORTA, R20
  ;Write Data to PORTA
  Here: rjmp Here
  
```



# IMDL Software Series Motor Test 1



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

=====
Title: EGN-1935 Motor Test 1
Programmer: A. Antonio Arroyo
Date: September 24, 2013
Version: 1.0
Description:
  * The robot will wait 5 seconds and then turns on both motors,
  * the left motor forward, the right motor in reverse. Toggles two
  * LEDs connected to Ports pins PA8 & PA1.
=====
#include "AAA.h"
//===== Prototypes =====
//===== Globals =====

int main(void) {
  int j=0, value=MAX_SPEED;

  init_system(); // Initialize Ports A, C, E
  init_motor(); // Initialize PPRC PWM on Timer 1
  init_lcd(); // set lcd in 4 bit mode, 2-line mode, with blinking cursor

  // 123456789ABCDEF
  lcd_string("EGN-1935 Robotic"); // Heading line 1 for LCD
  lcd_row(1);
  // 123456789ABCDEF
  lcd_string("IMDL Motor Tst 1"); // Heading line 2 for LCD

  // Delay a few seconds -5s
  delay(5000);

  lcd_clear();
  lcd_string("Left/Right Motor");

  for (j = 0; j = 180; j++) { // finite loop
    motor(LEFT_MOTOR, value);
    motor(RIGHT_MOTOR, -value);

    // Update LCD Panel
    lcd_row(1);
    lcd_string("L:");
    lcd_int(value);
    lcd_string("R:");
    lcd_int(-value);
    lcd_string(" ");
    lcd_int(j);

    // complement pin PA8-PA1 and write back
    if (PINA & 0x01) (PORTA^PINA)&=PE; PORTA^PINA|=0x02;
    else (PORTA^PINA)&=PF; PORTA^PINA|=0x01;

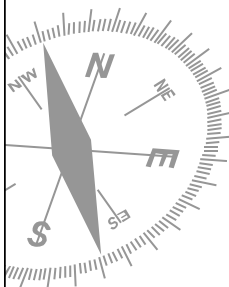
    // Delay about ~100ms
    delay(100);
  } // end for loop

  PORTA^PINA|=0x03; // Turn on both LEDs to end
  motor(LEFT_MOTOR, 0); // Stop the left motor
  motor(RIGHT_MOTOR, 0); // Stop the right motor
  return 0;
} // end main
=====

```



# IMDL Software Series Motor Test 2



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

=====
Title: EGN-1935 Motor Test 2
Programmer: A. Antonio Arroyo
Date: September 24, 2013
Version: 1.0
Description:
  * The robot will wait 5 seconds and then turns on both motors,
  * the right motor forward, the left motor in reverse.
  * The speed is obtained from a pot (input connected to ADC channel A3)
  * Also Toggles two LEDs connected to Ports pins PA8 & PA1.
=====
#include "AAA.h"
//===== Prototypes =====
//===== Globals =====
void sensor_module();
int pot = 0; // value for the potentiometer on ADC A3

int main(void) {
  int speed, speed1, value;

  init_system(); // Initialize Ports A, C, E
  init_motor(); // Initialize PPRC PWM on Timer 1
  init_lcd(); // set lcd in 4 bit mode, 2-line mode, with blinking cursor
  init_analog(); // Initialize ADC System Analog Channels A0-A3

  // 123456789ABCDEF
  lcd_string("EGN-1935 Robotic"); // Heading line 1 for LCD
  lcd_row(1);
  // 123456789ABCDEF
  lcd_string("IMDL Motor Tst 2"); // Heading line 2 for LCD

  // Delay a few seconds -5s
  delay(5000);
  lcd_clear();
  lcd_string("Varying Speed");

  for(j=0; j=180; j++) { // finite loop
    sensor_module();
    value = pot;
    value = MIN(value,1000) / 10;
    speed = value;
    speed1 = -value;
    motor(LEFT_MOTOR, speed);
    motor(RIGHT_MOTOR, speed1);
    // Display values on the LCD
    lcd_clear();
    lcd_string("ADC3:");
    lcd_int(pot);
    lcd_row(1);
    lcd_string("L:");
    lcd_int(speed);
    lcd_string("R:");
    lcd_int(speed1);
    lcd_string(" ");
    lcd_int(j);

    // complement pin PA8-PA1 and write back
    if (PINA & 0x01) (PORTA^PINA)&=PE; PORTA^PINA|=0x02;
    else (PORTA^PINA)&=PF; PORTA^PINA|=0x01;

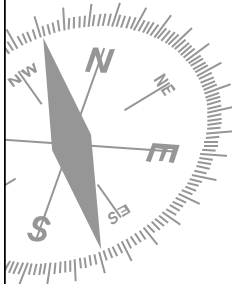
    // Delay about ~100ms
    delay(100);
  } // end for loop

  PORTA^PINA|=0x03; // Turn on both LEDs to end
  motor(LEFT_MOTOR, 0); // Stop the left Motor
  motor(RIGHT_MOTOR, 0); // Stop the Right Motor
  return 0;
} // end main
void sensor_module() { // Collect input sensor data
  pot = analog(3);
}
=====

```



# IMDL Software Series Hacked Servo Test



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

hst.c:59  main()
-----
Title:      EGN-1935 Robots Hacked Servo Test
Programmer: A. Antonio Arroyo
Date:      September 24, 2013
Version:   1.0
Description:
- Verifies the PWM system on the MIL Atmel Board on EGN-1935 Robot
- Is working by generating PWM signal on Timer1 using PFC PWM Mode
- On the EGN-1935 robots the left servo is connected to OC1A/PB5/PWA
- and the right servo is connected to OC1B/PB6/PWB
- Also toggles PAB & PA1 where LEDs w/ serial resistors are connected

Includes:
#include "Ada.h"

Prototypes:

Globals:

int main(void) {
    long i = 0;
    int value=1500;

    init_system(); // Initialize Ports A, C, E
    init_led(); // set led in 4 bit mode, 3-line mode, with cursor on and set to blink
    init_motor(); // Initialize Timer 1 PFC PWM System
    // 123456789ABCDEF
    lcd_string("EGN-1935 Robots"); // Heading Line 1 for LCD
    lcd_row(1);
    // 123456789ABCDEF
    lcd_string("Hacked Servo Test"); // Heading Line 2 for LCD
    // Delay a few seconds -ts
    delay(5000);
    lcd_clear();
    lcd_string("OC1A & OC1B");

    for (value = 500; value < 2501; value=value+50) { // finite loop
        OC1A=low; // One motor is wired in forward
        OC1B=high - value; // 2nd motor is wired in reverse
        // Display the Hacked Servo Values on the LCD
        // lcd_row(1);
        lcd_int(value);
        lcd_string(" ");
        lcd_int(3800 - value);
        lcd_string(" ");
        // Delay a few seconds(-ts) to read LCD
        delay(1000);
        // complement pin PAB-PA1 and write back
        if (PINA & 0x01) (PORTA^PINA)&=FE; PORTA^PINA|=0x02;
        else (PORTA^PINA)&=0x02; PORTA^PINA|=0x01;
    } // end for loop

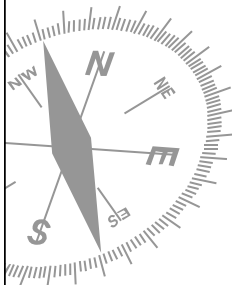
    PORTA^PINA|=0x03; // Turn on both LEDs to end
    OC1A=1500; // Stop Servo on PB5/OC1A
    OC1B=1500; // Stop Servo on PB6/OC1B
    return 0;
} // main

```

5



# IMDL Software Series IR & Cds Test



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

hst.c:64  sensor_module()
-----
Title:      EGN-1935 IRs & Cds Test
Programmer: A. Antonio Arroyo
Date:      September 24, 2013
Version:   1.0
Description:
- The robot will wait 5 seconds and then displays the IRs & Cds
- Input sensor values collected from ADC AB-A3
- Toggles two LEDs connected to Porta pins PAB & PA1

Includes:
#include "Ada.h"

Prototypes:
void sensor_module();

Globals:
int lir, cdir, rdir, cds;

int main(void) {
    long j=0;
    init_system(); // Initialize Ports A, C, E
    init_motor(); // Initialize PFC PWM on Timer 1
    init_led(); // set led in 4 bit mode, 2-line mode, with blinking cursor
    init_analog(); // Initialize ADC System Analog Channels AB-A3
    // 123456789ABCDEF
    lcd_string("EGN-1935 Robots"); // Heading Line 1 for LCD
    lcd_row(1);
    // 123456789ABCDEF
    lcd_string("IMDL IR/Cds Test"); // Heading Line 2 for LCD
    // Delay a few seconds -ts
    delay(5000);

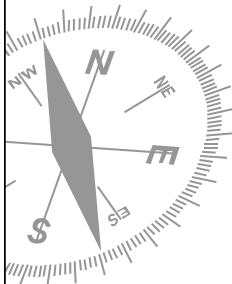
    for (j = 0; j < 100; j++) { // finite loop
        sensor_module();
        // Display sensor values from ADC AB-A3
        lcd_clear();
        // 123456789ABCDEF
        lcd_string("  C  R  Cd ");
        lcd_int(lir);
        lcd_row(1);
        lcd_int(cdir);
        lcd_string(" ");
        lcd_int(rdir);
        lcd_string(" ");
        lcd_int(cds);
        lcd_string(" ");
        // complement pin PAB-PA1 and write back
        if (PINA & 0x01) (PORTA^PINA)&=FE; PORTA^PINA|=0x02;
        else (PORTA^PINA)&=0x02; PORTA^PINA|=0x01;
        // Delay about -100ms
        delay(100);
    } // end for loop
    PORTA^PINA|=0x03; // Turn on both LEDs to end
    return 0;
} // main
void sensor_module() { // Collect input sensor data
    lir = analog(0);
    cdir = analog(1);
    rdir = analog(2);
    cds = analog(3);
}

```

6



# IMDL Software Series Obstacle Avoidance



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

=====
*
* Title: EGN-1935 IR-based Obstacle Avoidance
* Programmer: A. Antonio Arroyo
* Date: September 24, 2013
* Version: 1.0
*
* Description:
* The robot will wait 5 seconds and then move forward & read IRs
* It will then check the IRs and decide whether to turn or not
=====
#include "AAA.h"
===== Prototypes =====
void turn();
void sensor_module();
===== Globals =====
int lir, cir, rir, cds;
int main(void) {
  loop j=0;
  int speedr, speedl;
  init_system(); // Initialize Ports A, C, E
  init_lcd(); // set lcd in 4 bit mode, 2-Line mode, with cursor on and set to blink
  init_motor(); // Initialize Timer 1 PFC PWM System
  init_analog(); // Initialize ADC : Ports A0-A3
  lcd_clear(); // Clear LCD Panel
  // 123456789ABCDEF
  lcd_string("EGN-1935 Robots"); // Heading line 1 for LCD
  lcd_row(1);
  // 123456789ABCDEF
  lcd_string("Obstacle Avoid."); // Heading line 2 for LCD
  // Delay a few seconds --5s
  delay(5000);

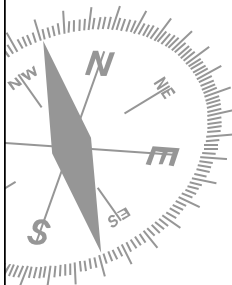
  for (j = 0; j = 100; j++) { // finite loop
    sensor_module();
    // Display Sensor Values on Line 0
    lcd_clear();
    lcd_int(lir);
    lcd_string(" ");
    lcd_int(cir);
    lcd_string(" ");
    lcd_int(rir);
    lcd_string(" ");
    lcd_int(cds);
    lcd_string(" ");
    if (lir > THRESHOLD) // Veer Right
      speedr = -MAX_SPEED;
    else
      speedr = MAX_SPEED; // Go straight
    if (rir > THRESHOLD) // Veer Left
      speedl = -MAX_SPEED;
    else
      speedl = MAX_SPEED; // Go straight
    motor(LEFT_MOTOR, speedl);
    motor(RIGHT_MOTOR, speedr);
    // Display the (Left,Right) Motor Speed on Line 1
  }
}
=====

```

7



# IMDL Software Series Obstacle Avoidance



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

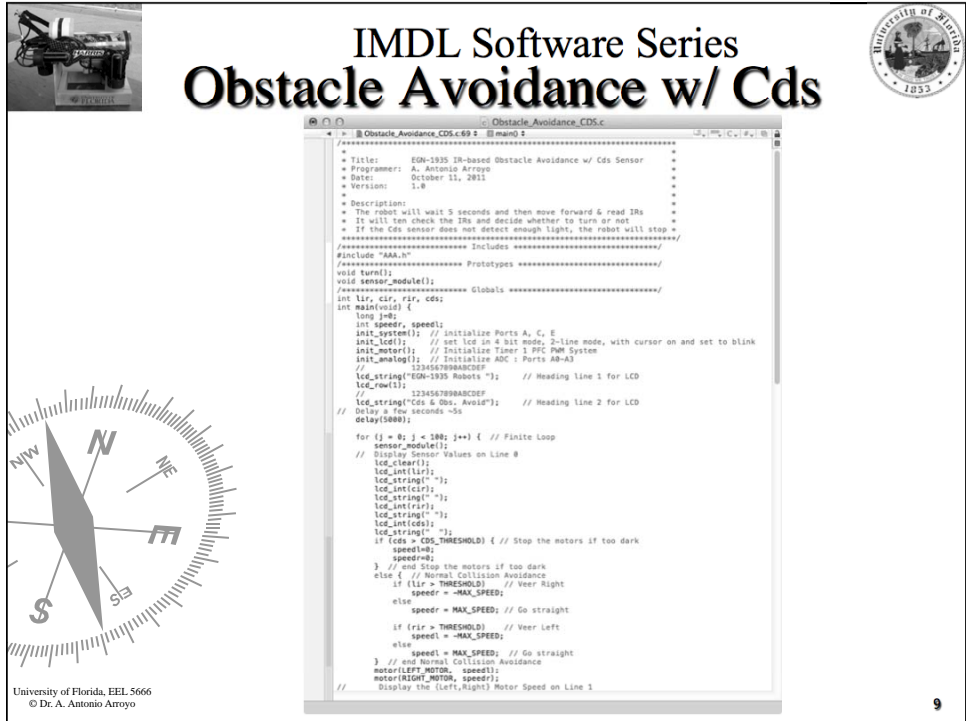
```

=====
*
* Title: EGN-1935 IR-based Obstacle Avoidance
* Programmer: A. Antonio Arroyo
* Date: September 24, 2013
* Version: 1.0
*
* Description:
* The robot will wait 5 seconds and then move forward & read IRs
* It will then check the IRs and decide whether to turn or not
=====
#include "AAA.h"
===== Prototypes =====
void turn();
void sensor_module();
===== Globals =====
int lir, cir, rir, cds;
int main(void) {
  loop j=0;
  int speedr, speedl;
  init_system(); // Initialize Ports A, C, E
  init_lcd(); // set lcd in 4 bit mode, 2-Line mode, with cursor on and set to blink
  init_motor(); // Initialize Timer 1 PFC PWM System
  init_analog(); // Initialize ADC : Ports A0-A3
  lcd_clear(); // Clear LCD Panel
  // 123456789ABCDEF
  lcd_string("EGN-1935 Robots"); // Heading line 1 for LCD
  lcd_row(1);
  // 123456789ABCDEF
  lcd_string("Obstacle Avoid."); // Heading line 2 for LCD
  // Delay a few seconds --5s
  delay(5000);

  for (j = 0; j = 100; j++) { // finite loop
    sensor_module();
    // Display Sensor Values on Line 0
    lcd_clear();
    lcd_int(lir);
    lcd_string(" ");
    lcd_int(cir);
    lcd_string(" ");
    lcd_int(rir);
    lcd_string(" ");
    lcd_int(cds);
    lcd_string(" ");
    if (lir > THRESHOLD) // Veer Right
      speedr = -MAX_SPEED;
    else
      speedr = MAX_SPEED; // Go straight
    if (rir > THRESHOLD) // Veer Left
      speedl = -MAX_SPEED;
    else
      speedl = MAX_SPEED; // Go straight
    motor(LEFT_MOTOR, speedl);
    motor(RIGHT_MOTOR, speedr);
    // Display the (Left,Right) Motor Speed on Line 1
  }
}
=====

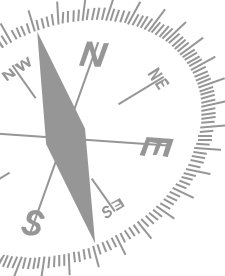

```

8



**IMDL Software Series**  
**Obstacle Avoidance w/ Cds**

University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

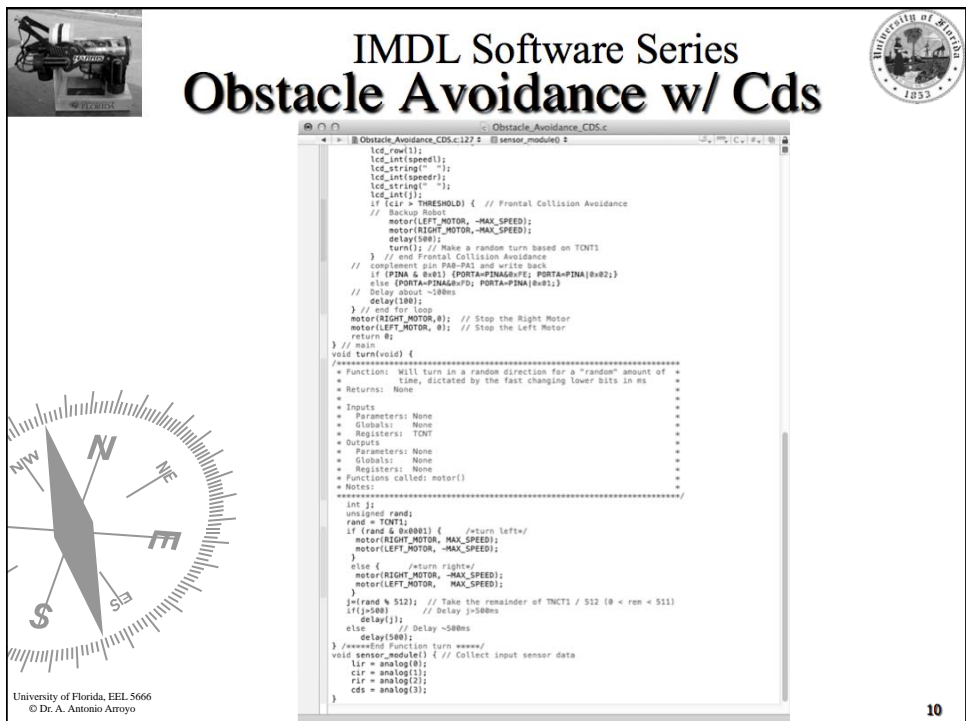



```

Obstacle_Avoidance_CDS.c:69:1: main1
/*****
 *
 * Title: EGU-1935 IR-based Obstacle Avoidance w/ Cds Sensor
 * Programmer: A. Antonio Arroyo
 * Date: October 11, 2011
 * Version: 1.0
 *
 * Description:
 * The robot will wait 5 seconds and then move forward & read IRs
 * It will then check the IRs and decide whether to turn or not
 * If the Cds sensor does not detect enough Light, the robot will stop
 *****/
#include "AA.h"
/***** Prototypes *****/
void turn();
void sensor_module();
int lir, ccr, rir, cds;
int main(void) {
    long j=0;
    int speedr, speedl;
    init_system(); // Initialize Ports A, C, E
    init_lcd(); // set lcd in 4 bit mode, 2-line mode, with cursor on and set to blink
    init_motor(); // Initialize Timer 1 SPC PWM System
    init_module(); // Initialize ADC: Ports A0-A3
    // 123456789ABCDEF
    lcd_string("EGU-1935 Robots"); // Heading line 1 for LCD
    lcd_row(1);
    // 123456789ABCDEF
    lcd_string("Cds & Obs. Avoid"); // Heading line 2 for LCD
    // Delay a few seconds -5s
    delay(5000);
    for (j = 0; j < 100; j++) { // Finite Loop
        sensor_module();
        // Display Sensor Values on Line 0
        lcd_clear();
        lcd_int(lir);
        lcd_string(" ");
        lcd_int(ccr);
        lcd_string(" ");
        lcd_int(rir);
        lcd_string(" ");
        lcd_int(cds);
        lcd_string(" ");
        if (cds > CDS_THRESHOLD) { // Stop the motors if too dark
            speedr=0;
            speedl=0;
        } // end Stop the motors if too dark
        else { // Normal Collision Avoidance
            if (lir > THRESHOLD) // Veer Right
                speedr = -MAX_SPEED;
            else
                speedr = MAX_SPEED; // Go straight
            if (rir > THRESHOLD) // Veer Left
                speedl = -MAX_SPEED;
            else
                speedl = MAX_SPEED; // Go straight
        } // end Normal Collision Avoidance
        motor(LEFT_MOTOR, speedl);
        motor(RIGHT_MOTOR, speedr);
        // Display the (Left,Right) Motor Speed on Line 1
    }
}

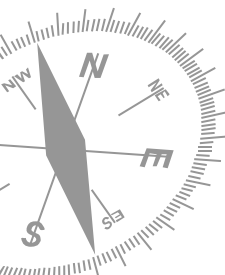

```

9



**IMDL Software Series**  
**Obstacle Avoidance w/ Cds**

University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo

```

Obstacle_Avoidance_CDS.c:127:5: sensor_module
    lcd_row(1);
    lcd_int(speedl);
    lcd_string(" ");
    lcd_int(speedr);
    lcd_string(" ");
    lcd_int(j);
    if (lir > THRESHOLD) { // Frontal Collision Avoidance
        // Backup Robot
        motor(LEFT_MOTOR, -MAX_SPEED);
        motor(RIGHT_MOTOR, -MAX_SPEED);
        delay(500);
        turn(); // Make a random turn based on TCNT1
    } // end Frontal Collision Avoidance
    // complement pin PA0-PA3 and write back
    if ((PINA & 0x01) & (PORTA & PINA) & 0x01) PORTA = PINA;
    else ((PINA & 0x01) & (PORTA & PINA) & 0x01) PORTA = ~PINA;
    // Delay about -100ms
    delay(100);
    // end for loop
    motor(RIGHT_MOTOR, 0); // Stop the Right Motor
    motor(LEFT_MOTOR, 0); // Stop the Left Motor
    return 0;
} // main
void turn(void) {
/*****
 * Function: Will turn in a random direction for a "random" amount of
 * Time, dictated by the fast changing lower bits in ms
 * Returns: None
 *
 * Inputs:
 * Parameters: None
 * Globals: None
 * Registers: TCNT1
 * Outputs:
 * Parameters: None
 * Globals: None
 * Registers: None
 * Functions called: motor()
 * Notes:
 *****/
    int i;
    unsigned rand;
    rand = TCNT1;
    if (rand & 0x0001) { //return left+
        motor(RIGHT_MOTOR, MAX_SPEED);
        motor(LEFT_MOTOR, -MAX_SPEED);
    }
    else { //return right+
        motor(RIGHT_MOTOR, -MAX_SPEED);
        motor(LEFT_MOTOR, MAX_SPEED);
    }
    j=(rand % 512); // Take the remainder of TCNT1 / 512 (0 < rem < 511)
    if(j>500) // Delay 3-500ms
        delay(1);
    else // Delay -500ms
        delay(500);
} //====End Function turn====
void sensor_module() { // Collect input sensor data
    lir = analog(0);
    ccr = analog(1);
    rir = analog(2);
    cds = analog(3);
}

```

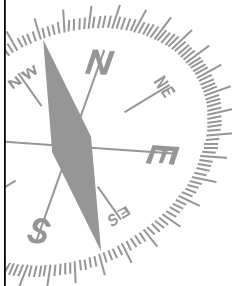
10



## IMDL Software Series



# *The End!*



University of Florida, EEL 5666  
© Dr. A. Antonio Arroyo