SAN JOSE STATE UNIVERSITY

Electrical Engineering

Solar Powered Automatic Lawn Mower
"Lawn Buddy"

Darwin Ramos

Jessie Lucero

Advisor: Dr. Kwok

December 9, 2009

Overview

Features:

- Three safety sensors
    - PIR Sensor (Human detection)
    - Ultrasonic Sensor (Object detection)
    - Accelerometer (prevents lawn operations while being held)
- Mulching cutting system
    - Cuts grass into tiny pieces that will later fertilize the lawn
- Solar powered
- “Plug and Cut” Design
    - No installation required

Benefits:

- Zero emissions
- Helps reduce the 5% of the U.S. pollution caused by gas powered lawn mowers
- Will not show up on electrical bill

Description

This project is an autonomous lawn mower that will allow the user to the ability to cut their grass with minimal effort. Unlike other robotic lawn mowers on the market, this design requires no perimeter wires to maintain the robot within the lawn. Through an array of sensors, this robot will not only stay on the lawn, it will avoid and detect objects and humans. This design is still in the prototype stage due to financial and time constraints. Documentation includes all major design aspects. This project will continue in hopes to market the design.

Introduction

In the time where technology is merging with environmental awareness, consumers are looking for ways to contribute to the relief of their own carbon footprints. Pollution is man made and can be seen in our own daily lives, more specifically in our own homes. Gas powered lawn mower are in 90% of U.S. home and they create 5% of the total U.S. pollution. Green technology initiatives are being support by both the government and cooperates business. Our new design for an old and outdated habit will help both the consumer and the environment.

This project of a solar powered automatic lawn mower will relieve the consumer from mowing their own lawns and will reduce both environmental and noise pollution. This design is meant to be an alternate green option to the popular and environmentally hazardous gas powered lawn mower. Ultimately, the consumer will be doing more for the environment while doing less work in their daily lives. The hope is to keep working on this project until a suitable design can be implemented and then be ultimately placed on the market.

Overview

This design contains a microcontroller, multiple sensors, and a solar charging system. Adding these elements together, we get our robotic lawn mower. The sensors are the eyes of our robot. The goal was to let our robot see the difference between grass and concrete while monitoring its surroundings continuously. Initially, we had an idea what type of sensors we wanted to use. Our robot needed to detect if it was on grass versus on concrete and we decided to use a humidity sensor. Since concrete/dirt and grass are distinctively different in density and moisture levels, the humidity was a good factor to distinguish both materials. In addition to sensing humidity, we wanted object detection; both humans and objects. In which case, we went with using a passive infra red sensor to detect the heat radiation from humans and an ultrasonic sensor to detect if the robot was heading into an object. Safety is the main concern when designing a robot with blades. We wanted our robot not to start operating if it was being held in the air by the user. Knowing that the user would be randomly holding the robot we needed a sensor to detect orientation. The accelerometer was thought of being used because it can detect its orientation based on pre calibrated axis orientation. The power the system there are many options. With recharging batteries, there are various chemistries but we decided to go with the one that work best with solar charging. The nickel-metal hydride (NiMH) was found to be the best because given a low charging current, it will not over charge. Sizing the battery will depend on what we are powering, specifically the motors. Like batteries, there is a range of motors to choose from. We went with two 7.2 DC motors with integrated gear heads. The needed torque did not need to be a lot because we were going to have a small prototype. These motors have 100 oz-in torque which is plenty for our design. The block diagram of our design is shown in figure 1.

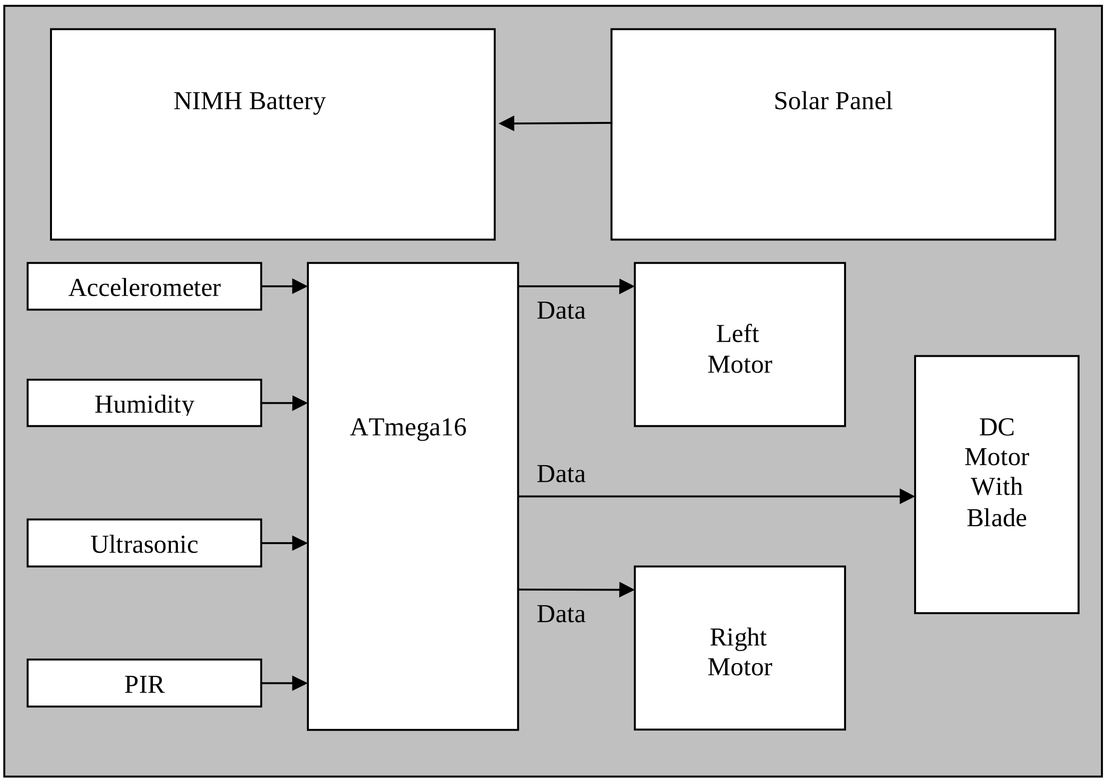

FIGURE 1- System Block Diagram

Determining where to place our sensors is crucial to the overall effectiveness of our design. Initially, we knew to place the humidity sensor facing down into the ground. The solar panels were to be placed horizontal on the robot because to achieve maximum sun exposure. The microprocessor must be in the robot to protect it from the natural elements. Our ultrasonic sensor will be mounted directly in front of the robot for maximum detection. The only sensor that will be angled is the PIR because it needs to detect humans and since the robot is at ground level it must be facing up to effectively detect humans. Our preliminary design is shown in figure 2.

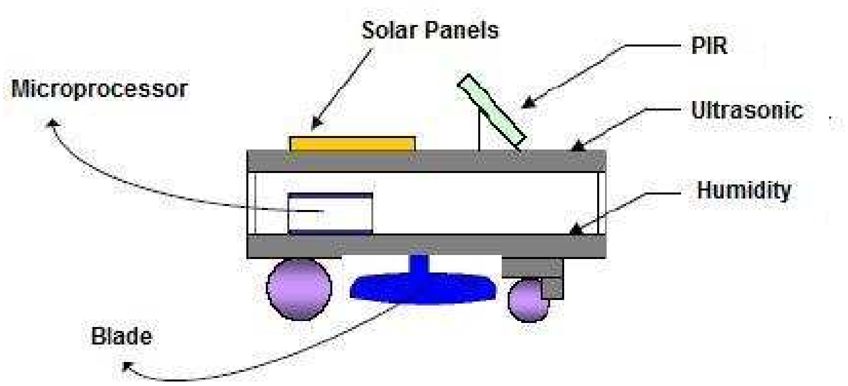

FIGURE 2– Preliminary Design

Components:

The main factor in deciding which components to use was that it must be low power high efficiency. It was preferred that every sensor had a break away board because it would make interfacing much more easily. The costs went up but overall stability was achieved.

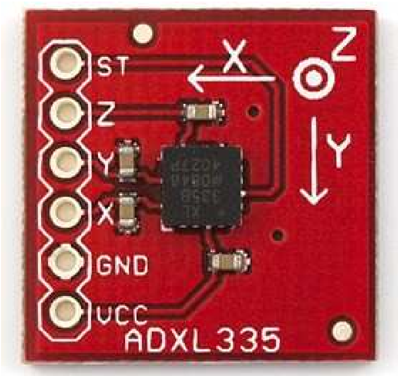

FIGURE 3-Three Axis Micromachined Accelerometer

The **ADXL335** from **Analog Devices** is an accelerometer that has 3 axis in which it can be measured. Essentially, this chip outputs analog signals which correspond to the orientation of the chip. Each axis has its own pin. Having the chip rest on its back produced a signal that is 50% of the supplied voltage (2.5V). If the chip moved, the voltage increased in it went into positive direction of the axis (printed on board) and decreases when oriented into the negative orientation. This chip has six pins: Vcc (5V), ground, self test (not used), and three pins for the three axis orientation.

FIGURE 4-PIR Motion Sensor

The **RB-Plx-75** from **Parallax** is a PIR sensor was chosen based on short range abilities. If the robot detects a human close up then it will stop what ever it is doing until the human leaves the area. This specific PIR sensor will output a pulse of 5V if it detects humans and will output 0V during idle. The advantage of this sensor is that it will reset itself if when it detects nothing. But if it detects something multiple times then the output will stay logic high (5V) until the human leaves the area. This feature was used in our design to let our robot know the difference between someone who is just passing by and someone else who is staying too long in the area, which is a dangerous situation. This chip has three pins: Vcc (5V), ground, and ‘alarm’ data pin.

FIGURE 5 -High Performance Sonar Module

The **LV-MaxSonar-EZ1** from **Maxbotix** is a high performance sonar module that can detect an object in its ‘vision’ from 20 feet away. In this design we will want to detect objects that are in the path of the lawn mower. We decided that detection should start at 3 inches from the robot. This sensor produces an analog voltage proportional to the detected object distance. When powered at 5V, this sensor outputs 10mV per inch of object detection distance. There are seven pins but we are only using three (Vcc, ground, and data). Since we are measuring low signals we decided to use a low pass filter to clean up the signal (figure 6).

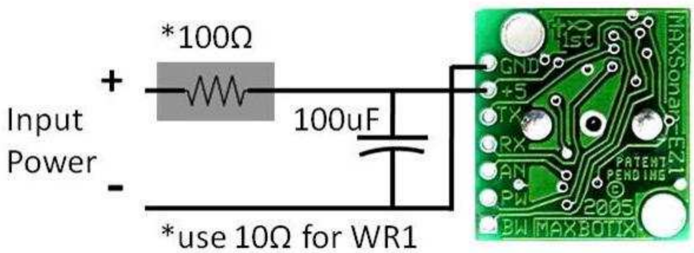

FIGURE 6 –Low Pass Filter

FIGURE 7 -Humidity Sensor

The **HIH-4000** from **Honeywell** was chosen due to the fast response time (5us). Simply put, this sensor outputs an analog signal linearly proportional to he relative humidity that it measures: the higher the humidity the higher the output voltage (figure 8). There were other humidity sensors with different interfaces (I2C and variable capacitance) but they were not chosen due to efficiency and response time. This sensor has three pins: Vcc (5V), ground, and data.

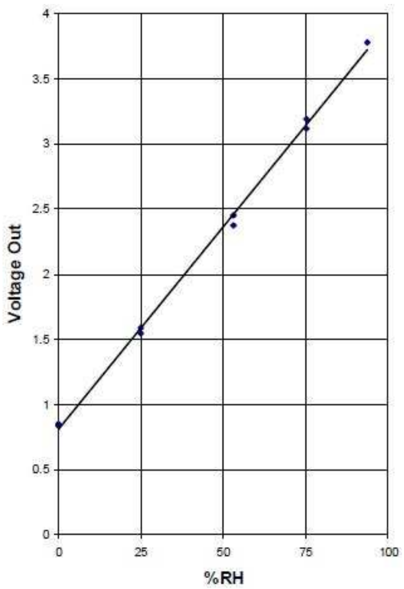

FIGURE 8 –Plot of Relative humidity versus output voltage

FIGURE 9 -Quadruple Half-Driver

To control the DC motors, a H-bridge was used for both microcontroller protection and efficiency. Even though each chip can handle two bi-directional DC motors, we went with one chip per motor for heating concerns. Figure 10 shows the schematic used for each motor.

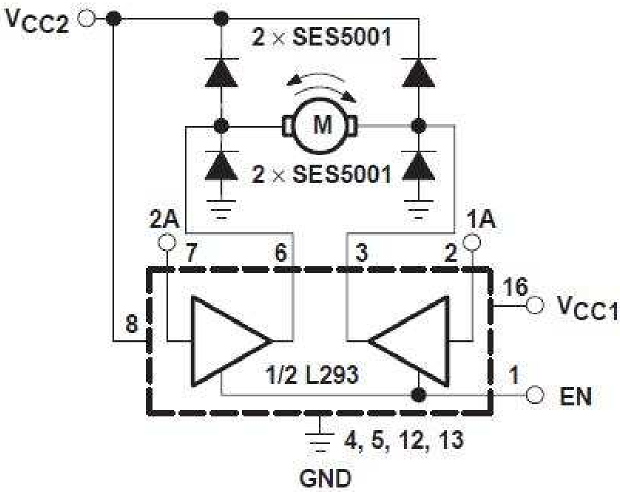

FIGURE 10 –Bi-directional motor control with L293DNE

The L293DNE was chosen over the L293NE because the diodes were internally integrated. The pin assignments are: pin 8 to 9.6V, pin 16 to 5V, pin 4,5,12,13 to ground, pin 1 as enable, and pin 7 and 2 are the input pins. The input pins (1A and 2A) have logic assignments shown in figure 11.

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

FIGURE 11 –Logic table for L293DNE

FIGURE 12 -Battery NiMH

This battery was chosen to fit our power needs at 9.6V (2100mAh). We chose the chemistry to be nickel metal hydride because it works perfectly with solar power and our solar charging system.

FIGURE 13 -Solar Panel

This solar panel from spark fun electronics is rated at 8V open voltage and 310mA short circuit. Keeping in mid that the current rate is not more than 10% of the current charge of the battery (2100mAh) we will not damage the battery due to over charging.

FIGURE 14 -ATmega16 Microcontroller

For any robotic system, the microcontroller is the heart and it's where everything comes together. The ATmega16 has 8 ADC Channels, 4 PWM Channels, 16K Bytes Flash, Low Power, and 32 I/O lines. The ADC channels are being used for the analog to digital conversions from our sensors, four PWM channels which will be used on the enable pins of our H-Bridge which controls the speed of our wheels, and 4 output pins will be used for the DC motors (2 per motor).

Cutting Patterns

The lawn mower will have two types of cutting styles: spiral and random. The user will place the robot in the center of their lawn and let it cut. To achieve this cutting pattern both wheels must turn at two very different speeds with the outmost wheel moving the fastest. This can be done by applying two different PWM signals (explained later). Figure 15 shows the spiral pattern:

FIGURE 15- Spiral cutting pattern

Once the humidity sensor tells the microcontroller that it is on concrete then the robot will change cutting pattern and will randomly. Essentially, the robot will cut linearly until it is interrupted by the humidity sensor or the ultrasonic sensor. If it is interrupted by the ultrasonic sensor then that means there is an object in its path and will back up and turn right. In the case that the humidity sensor measures low humidity (on concrete) it will also stop and back up and turn right. Theoretically if the robot has enough power it can cut the entire lawn. Figure 16 shows the random cutting pattern:

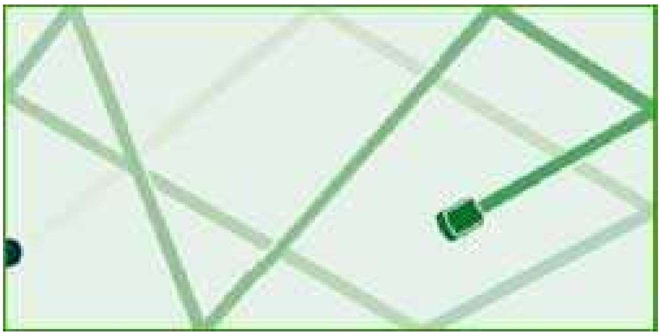

FIGURE 16- Random cutting pattern

Testing

Using the datasheets from each sensor and the microcontroller we constructed the circuit shown in figure 15. The L293DNE H-Bridge could support up to two bidirectional DC motors but to minimize heat for each chip we had one chip for each motor. When the DC motor is stopped suddenly a back induced electromagnetic field is produced and can damage the H-Bridge. To reduce this effect four diodes are placed in opposite directions (figure 10). The L293DNE has these diodes internally which reduces clutter in our circuit. The trade off is that the current supplied to the motors is reduced. Each sensor has an analog output and it is being inputted in to the Analog-to Digital converter (ADC) of the microcontroller.

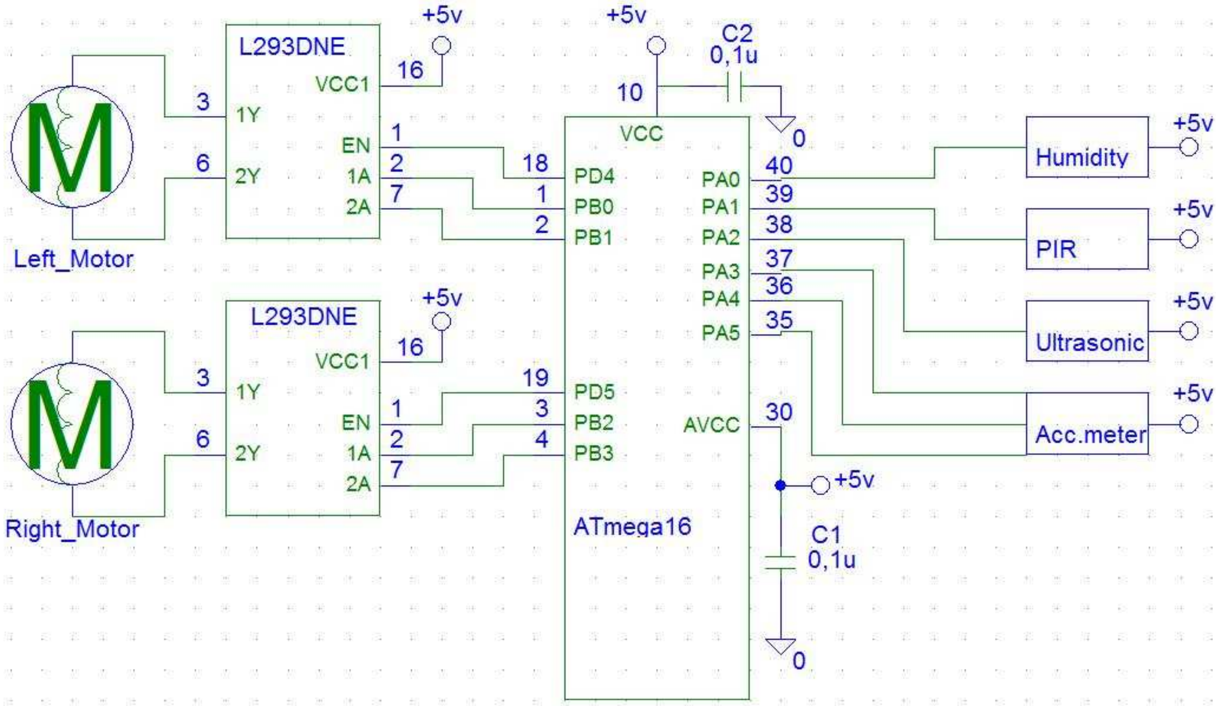

FIGURE 17 – S.P.A.L.M. Schematic

The Atmega16 has 4 Pulse Width Modulation (PWM) channels, we used only 2: one for each motor. The PWM will control the speed of each motor. Essentially, the PWM is a square function with a DC offset that repeats every cycle. 'The 'ON' time determines how much of the voltage is being applied to the motors. The higher the voltage the faster the motor moves. Figure 16 shows two different PWM signals. The top PWM signal will move the motors slower than the lower PWM signal.

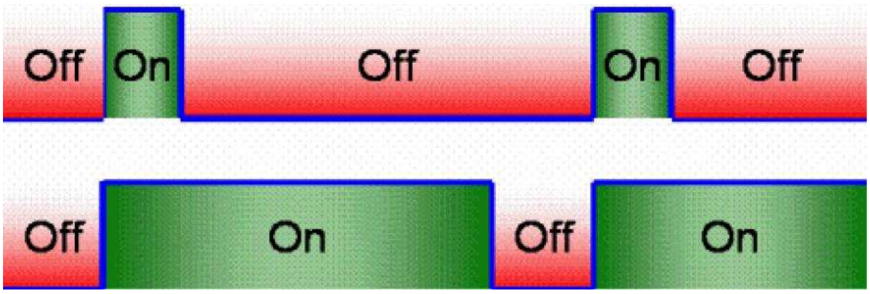

FIGURE 18– Two PWM Signals

The speed of the robot must not be too fast because the sensors need time to measure its environment. There are three modes of PWM and after some research the Phase correct PWM is the best for DC motors. Figure 17 shows the PWM of our robot (50% duty cycle).

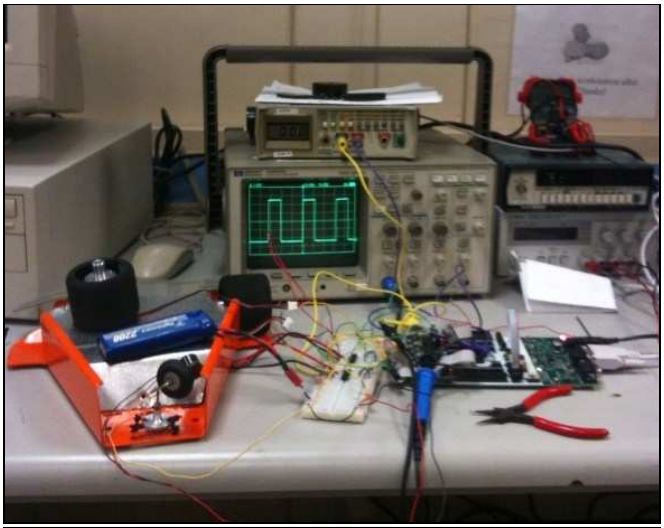

FIGURE 19–Testing PWM

The microcontroller code in ‘C’ is shown in the comment box below. The code comes with comments that will explain which registers are being set to produce the PWM signal.

```
#include <avr/io.h>
#include <util/delay.h>
#include "uart.h"

void init();
void go_left();
void go_right();
void  spiral();
void random();
void backup();
void pwm();

int  isSomethinginproximity();
int  isSomethingInMyWay();
int  amIonconcrete();

int main()
{
       init();                                          //Calls initialize
function

while(1)
```

```
{
spiral();                                                      //Calls spiral function

if( amIonconcrete())                      //If detect concrete
        {
                backup();                                //Calls backs up
function
                _delay_ms(3000);                 //delay 3 seconds
               go_right();                               // Calls go right
function
               pwm();                                    //Calls and sets PWM
signal
               PORTB = 0xFA;                     //Forward
        }
if ( isSomethingInMyWay())               //If detects object
        {
               stop();                                   //Calls stop function
               _delay_ms(5000);                  //delay 5 seconds
               backup();                                 //Calls back up function
               go_right();
               pwm();                                    //Calls go right
function
               PORTB = 0xFA;                     //Forward
}
}
}
void init()
{
        //Clock prescaling 256
        // Select 8 bit phase correct with TOP=255
        TCCR1A = _BV(WGM10) | _BV(COM1A1) | _BV(COM1B1);     //sets PWM
signal output to PD5


        TCCR1B =   _BV(CS12);                    //sets PWM signal output to PD4

        DDRD |= _BV(PD4) | _BV(PD5);  //sets PD4 and PD5 as output

        DDRA = 0x00;                             //Sets PORTA as input for ADC
convertion
        PORTA = 0x00;

        DDRB = 0xFF;                             //Sets PORTB as input for motors
        PORTB = 0x00;

        uart_init();                             //initialize uart for
hyperterminal
        printf("STARTING !\n");
}
void go_left()
{
        pwm();
    PORTB =0xF2;                //only the right wheel moves forward
        _delay_ms(1000);        //delay 1 second
        printf("go_left \n");
}
void go_right()
```

```
{
        pwm();
    PORTB  =0xF8;              //only the left wheel moves forward
        _delay_ms(1000);       //delay 1 second
        printf("go_right \n");
}
void spiral()
{
        pwm();
        PORTB = 0XFA;          //Move wheels move forward

        OCR1A = 0x80;          //Sets PWM signal for right motor to 50% duty
cycle
        OCR1B = 0;                     //Sets PWM signal for left motor to 0%
duty cycle
        printf("spiral() start\n");

int i=0 ;

while (!isSomethinginproximity() && !isSomethingInMyWay() &&
!amIonconcrete());
    {
     if (OCR1B <0x80)
         OCR1B = OCR1B + 1;    //PWM signal increases by 1% every .5 seconds
        _delay_ms(500);
    }
     printf("sprial() done\n");
        }
void backup()
{
    pwm();
        PORTB  =0xF5;          //Both wheels reverse
        printf("backup \n");

    _delay_ms(3000);   //delay 3 seconds
    pwm();
        PORTB =0xF1;           //turns right (only right wheel moving reverse)
        printf("reverse right \n");
      _delay_ms(1000);

    return;
}
int isSomethingInMyWay()
{
    uint16_t adc_ultrasonic;
    int retval;
        int i;

        ADMUX = _BV(REFS0) | _BV(MUX1); //Sets PA2 as ADC input for Ultrasonic
sensor
        ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS1) | _BV(ADPS0);
        ADCSRA |= _BV(ADSC);

        while (ADCSRA & _BV(ADSC))
                ;
        //do nothing while ADC is completed
```

```
        adc_ultrasonic = ADC;                 //Stores ADC value into
adc_ultrasonic
        printf("Ultrasonic ADC = %u\r\n ",adc_ultrasonic);

        for(i=0; i<10; i++)
                _delay_ms(10);

        if(adc_ultrasonic > 25) //if object is near return 1, if not return 0;
        {
                        retval = 1;
                        printf("isSomethingInMyWay returns %d\r\n", retval);
        }
        else retval = 0;
                printf("isSomethingInMyWay returns %d\r\n", retval);
    return retval;
}
int isSomethinginproximity()
{
        uint16_t adc_pir;
        int retval;
        int i;

        ADMUX = _BV(REFS0) | _BV(MUX0);       //Sets PA1 as ADC input for PIR
sensor
        ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS1) | _BV(ADPS0);
        ADCSRA |= _BV(ADSC);

        while (ADCSRA & _BV(ADSC) )
                ; //do nothing while ADC is completed
        adc_pir = ADC;                                         //Stores ADC
value into adc_pir
        printf("PIR ADC = %u\r\n ",adc_pir);

        for(i=0; i<10; i++)
                _delay_ms(10);

        if (adc_pir < 800)
                {
                         retval = 1;
                        printf("isSomethinginproximity returns %d\r\n",
retval);
        }
        else retval = 0;
                        printf("isSomethinginproximity returns %d\r\n",
retval);
        return retval;

}
int amIonconcrete()
{
                int retval;
        uint16_t adc_humidity;
                int i;
                 ADMUX = _BV(REFS0); //Sets PA0 as ADC input for humidity
sensor
                ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS1) | _BV(ADPS0);
                ADCSRA |= _BV(ADSC);
```

```
               while(ADCSRA & _BV(ADSC) )
                       ;                             //do nothing while ADC
conversion completes

               adc_humidity = ADC;    //stores ADC value into adc_humidity
               printf("Humidity ADC = %u\r\n ",adc_humidity);

               for(i=0; i<10; i++)
               _delay_ms(10);

            if (450>adc_humidity)     //detects if on concrete then return 1
else return 0;
                       {
                retval = 1;
                              printf("amIonconcrete returns %d\r\n", retval);
            }
                       else retval = 0;
                              printf("amIonconcrete returns %d\r\n", retval);
            return retval;
    }
void pwm()
{
OCR1A = 0x80;          //sets PWM to 50% duty cycle
OCR1B = 0x80;          //sets PWM to 50% duty cycle
}

void stop()
{
PORTC=0x00;                   //Stops motor
printf("stop \n");
}
```

FIGURE 20-Programming Code

<u>Summary</u>

Our design is meant to replace the gas powered lawn mower which 90% of U.S. homes own at least one. We believe that Lawn Buddy is the 21st century upgrade to the gas lawn mower. The user will turn on Lawn Buddy and place it in the center of the lawn and walk away while the lawn is being cut. There are two cutting patterns available for Lawn Buddy, spiral and random. Once Lawn Buddy is set in the center it will cut spirally (figure16) until one of the sensors is tripped. Once it senses it is not on grass any more (edge of the lawn) then it will back up and run the random cutting pattern (figure16). Safety features are embedded into the design. Such as: if the user hold onto Lawn Buddy too long after turning it on the robot will not start its programming until it is on ground level. This is achieved by the accelerometer that can detect the tilt motion in respect to the Earths gravity. Another safety feature is when the robot is cutting and someone (human or pet) gets in the way of the path of the robot it will stop momentarily. The PIR sensor allows us to detect humans. If it stops too many times then it will shut off completely. In addition to human detection it can also detect object that do not emit radiation heat such as pots and bushes. If any object is it its way it will stop and turn around. Unlike other robotic lawn mowers in the industry, this design will not require perimeter wires to keep the robot on the lawn. Instead it uses humidity sensors to detect if it is on grass (high relative humidity) or concrete (low relative humidity). Figure 18 shows various pictures of Lawn Buddy.

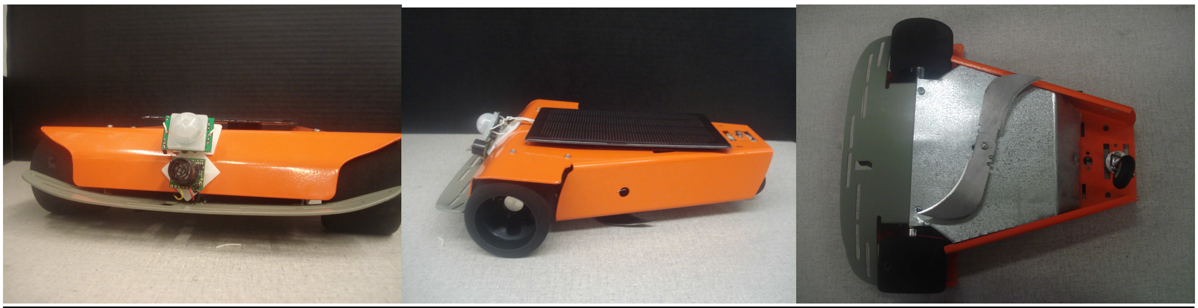

FIGURE 21- Profile pictures of Lawn Buddy

Bibliography

[1] Imran Ahmed, Hong Wong, and Vikram Sapila. (2004, July). Internet-Based Remote Control using a Microcontroller and an Embedded Ethemet. *Proceeding of the 2004 American Control conference*, 1329-1331. Retrieved on May 6, 2009 from IEEExplore

[2] Yan-Fang Li, Saul Harari, Hong Wong, and Vikram Kapila (2004, July). Matlab-Based Graphical User Interface Development for Basic Stamp 2 Microcontroller Projects. *Proceeding of the 2004 American Control conference*, 3233-3236. Retrieved on May 6, 2009 from IEEExplore

[3] Chia-Chang Tong (April 2005). The Development of Portable Infrared Color Sensor. 2-3. Retrieved on May 6, 2009 from IEEExplore

[4] M. B. Grier (September 2005). Infrared Color Translation. Proceedings of the Ire. 4 (3). 1574-1575. Retrieved on May 6, 2009 from IEEExplore

[5] Anshuman Panda, Hong Wong, Vikram Kapila, and Sang-Hoon Lee (December

2005). Matlab Data Acquisition and Control Toolbox for Basic Stamp Microcontrollers. *Proceedings of the 45th IEEE Conference on Decision & Control*. 3918-3925. Retrieved on May 6, 2009 from IEEExplore

[6] Yutaka Hiroi and Akinori Ito (June 2008). Are Bigger Robots Scary? The Relationship Between Robot Size and Psychological Threat. *International Conference on Advanced Intelligent Mechatronics*. 546-550. Retrieved on May 6, 2009 from IEEExplore

[7] R.Ramaprabha and B.L.Mathur (June 2008). Modelling and Simulation of Solar PV Array under Partial Shaded Conditions. *ICSET 2008*. 7-11. Retrieved on May 6, 2009 from IEEExplore

[8]Sung Jun Oh, Dong JOOn Ahn. and Ernest L. Hall (September 1989). A Wide Angle Vision System for Mobile Robots. *Center for Robotics Research*. 490-497. Retrieved on May 6, 2009 from IEEExplore

[9] Jason Smith, Scott Campbell, Jade Morton (July 2005). Design and Implementation of a Control Algorithm For an Autonomous Lawnmower. *Department of Electrical and Computer Engineering Miami University*. 456-459. Retrieved on May 6, 2009 from IEEExplore

[10] Haydar Sahin and Levent Guvenc (April 2007). Household Robotics Autonomous Devices for Vacuuming and Lawn Mowing. *IEEE Control Systems Magazine*. 20-24. Retrieved on May 6, 2009 from IEEExplore

[11] Taj Mohammad Baloch and Timothy Thien Ching Kae (July 2008). Design and Modelling a Prototype of a Robotic Lawn Mower. *Electrical and Electronic Engineering Department Universiti Teknologi PETRONAS*. 1-5. Retrieved on May 6, 2009 from IEEExplore

[12] Bing-Min Shiu and Chun-Liang Lin (March 2008). Design of an Autonomous Lawn Mower with Optimal Route Planning. *Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan*. 1-6. Retrieved on May 6, 2009 from IEEExplore

Appendix