STAN - The Smart Trash Can

Alisha Garrett
alishagarrett@hotmail.com
Department of Engineering Technology
University of Houston

Chris Vanderbles
cvanderbles@gmail.com
Department of Engineering Technology
University of Houston

Ricardo Moreno
ricdanmoreno@msn.com
Department of Engineering Technology
University of Houston

Farrokh Attarzadeh, Ph.D.
FAttarzadeh@uh.edu
Department of Engineering Technology
University of Houston

ABSTRACT

This paper presents the design and implementation of a system that promotes a cleaner and safer environment. STAN – The Smart Trash cAN, ensures safer and hygienic surroundings with the help of a number of sensors that include sensors which alert when the trash can is full or contains a high level of decomposing items and sensors to automatically open the lid when a patron is detected within a certain distance. The STAN core circuit consists of a Mini-Max/51C-2 microcontroller board manufactured by BiPOM Electronics [1] interfaced with an Infrared-Radio (IR) distance sensor [2], methane detector, force sensor [3] and other peripheral devices to communicate the status. The distance sensor was linked programmatically to the motor control relays to trigger opening and closing of the 'iris', a specially designed lid, when a user is detected at the trash can. The opening routine sets the direction of the motor and runs the motor until the "LidOpen" limit switch is activated. Once the "LidOpen" switch is activated, a timer is started to control how long the 'iris' mechanism will stay open before activating the closing routine. STAN. project was completed as part of the senior project requirements by a senior design student team consisting of three members under the supervision of the faculty and the graduate assistants.

INTRODUCTION

The major function of STAN is to assist in the janitorial duties of public, high volume areas and to contribute to the ultimate goal of a sanitary, germ-free environment. Therefore, STAN will be most useful in places such as food courts, schools, hospitals, entertainment venues, etc. which have the common issue of wastage control. Outside venues also encounter problems with rodents and insects. Trash cans tend to overflow and produce an unsanitary and unpleasant situation. Janitors can only monitor a particular trash can for so long before they have to go on and tend to other duties.

A super block that shows the inter-relation between the hardware and the software is shown in Figure 1.

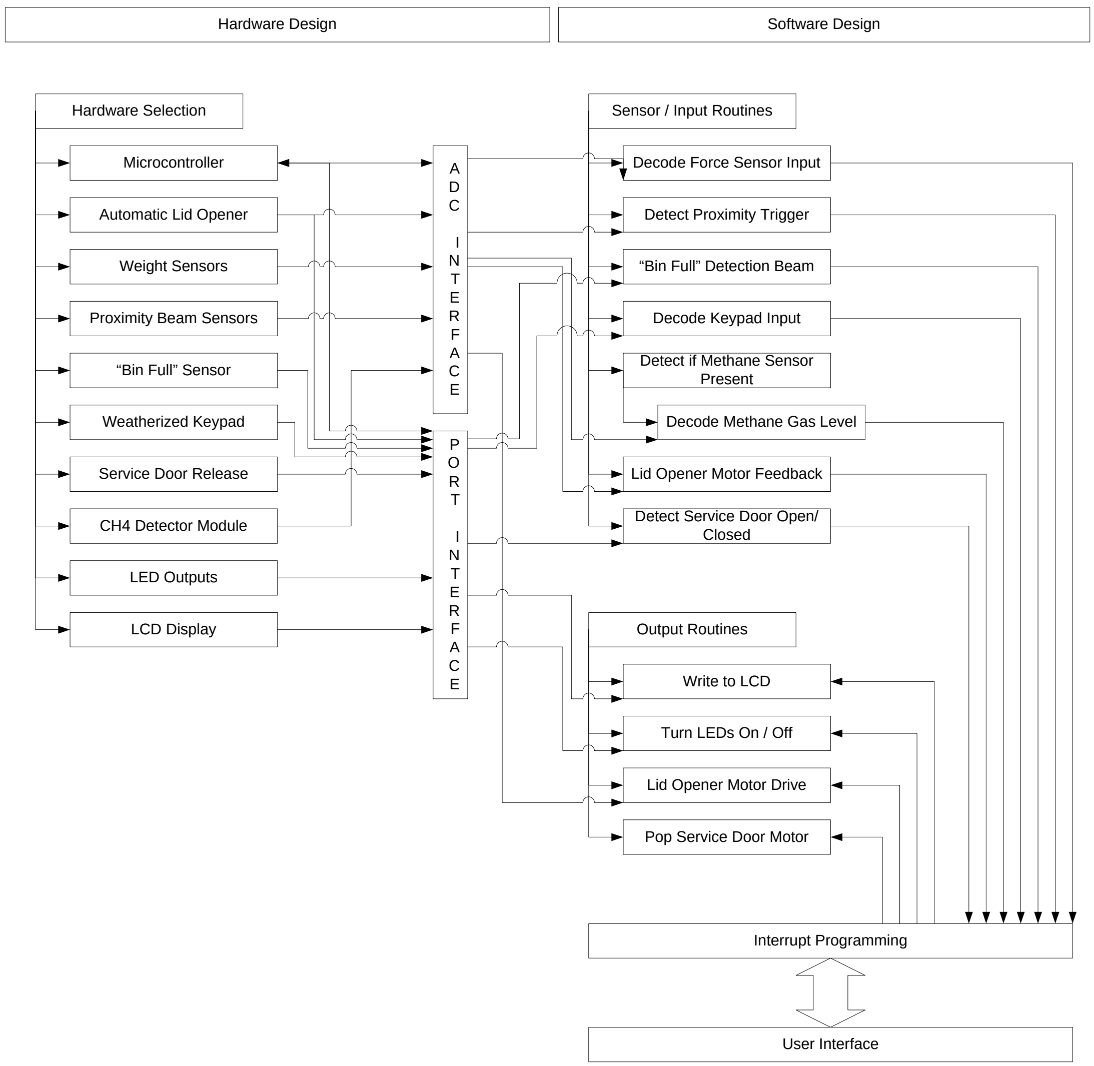

Figure 1. Super block

The super block diagram is divided into two main sections, with multiple points of interconnection. The hardware section depicts the hardware components of the system, as well as how they interface with the microcontroller. The components that interface with the analog to digital converter are the automatic lid opener, motor feedback component, weight/force sensors, proximity beam sensors, and an optional methane (CH_4) detector. The devices that interface directly to the microcontroller ports are the step motor driver for the lid opener, the "bin full" sensor, weatherized keypad, service door release motor, LED outputs, and LCD display. Each of the hardware devices is controlled by one or more corresponding software modules. Software modules are implemented with object-oriented programming design methodology. The sensor input routines consist of force sensor(s) input, proximity detector input, "bin full" input, keypad input, optional methane sensor input, lid opener motor feedback, and service door position switch input.

HARDWARE

This section describes how the sensors and the outputs are interfaced to the Mini-Max/51C-2 board with the help of relay circuits. The Hardware block diagram is shown in Figure 2.

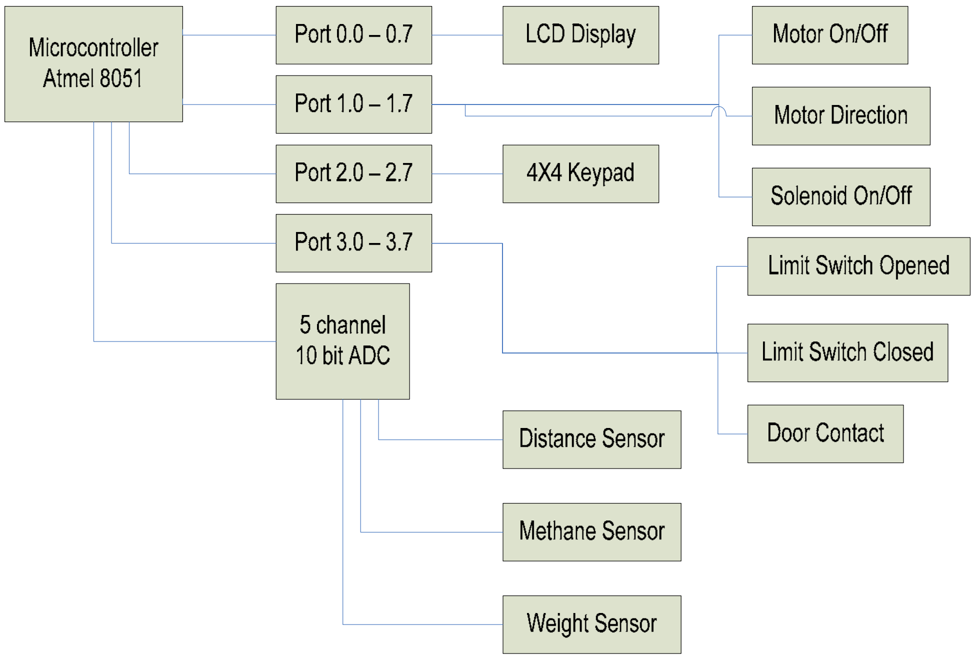

Figure 2. Hardware block diagram

The electrical hardware includes Mini-Max/51 C-2 microcontroller board, Figaro NGM2611-C13 [4] pre-calibrated methane sensor module based detector circuit is shown in Figure 3a, Tekscan FlexiForce® Sensor circuit, shown in Figure 3b, for detecting weight, Sharp GP2Y0A21YK based IR distance detector circuit shown in Figure 3c, 12 V and 5 V relays for interfacing the microcontroller, the power motor and the solenoid. Electronic construction methods included basic soldering and a small breadboard to construct the circuits. A standard 20-wire ribbon cable was used to expand the I/O bus instead of a peripheral board. The ribbon cable was chosen over the peripheral board for simplicity and low cost.

(a) (b) (c)

Figure 3. (a) methane sensor detector, (b) Tekscan FlexiForce® Sensor, (c) IR distance detector

The mechanical hardware includes the trash can designed to house all the circuitry, 'iris' blades, 'iris' base plate, and 'iris' actuator.(See Figures 4, 5, and 6).

Many of these components are physically and logically linked together. All three sensors interface to the ADC on the microcontroller. The methane sensor periodically checks the readings from the device and instructs the microcontroller according to the parameters set in the program. The weight sensor circuit monitors weight against a threshold level. The distance sensing circuit monitors for user presence in its vicinity and calls the motor opening and closing routine, which interfaces to both the motor on/off relay and the motor direction relay. The motor drives the actuating ring on the 'iris' mechanism, which in turn triggers the limit switches mounted adjacent to the actuating ring. The door contact is interfaced directly to a port on the microcontroller where its status is monitored and checked against a maintenance mode routine in the software to enable/disable the functionality of the trash can.

The LCD and keypad were attached to the microcontroller using the standard expansion cables provided on the microcontroller board.

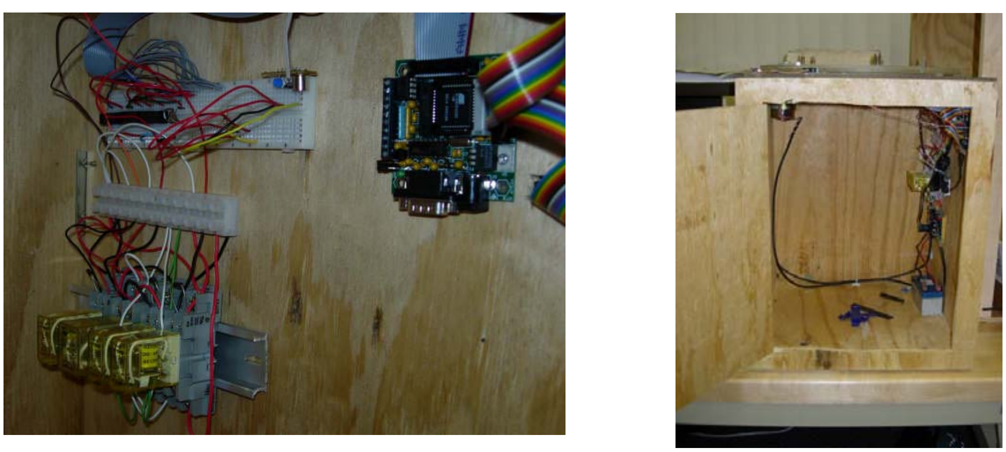

Figure 4. Components of STAN and interior view

The 'iris' mechanism was constructed using several components. The base plates and actuating rings were constructed from standard Plexiglas®. The actuating ring and the actuating ring guide are attached at several points. The base plate and the base ring are attached using the 5 holes that feed through for the blade pivot points. All holes are counter-sunk and connected with #6 flat head machine screws. The blades were constructed using standard sheet metal, and had their holes counter-sunk into the Plexiglas® to avoid collisions with other blades. The standoffs were connected to the blades using standard metal ½ inch standoffs, secured to the blades with #6 flat head machine screws. For assembly, the standoffs fit through the guides on the actuating ring and control all the blades in unison. The motor has a pulley attached to the drive shaft, which was then connected around the outside of the actuating ring; this allows the motor to drive the 'iris' mechanism.

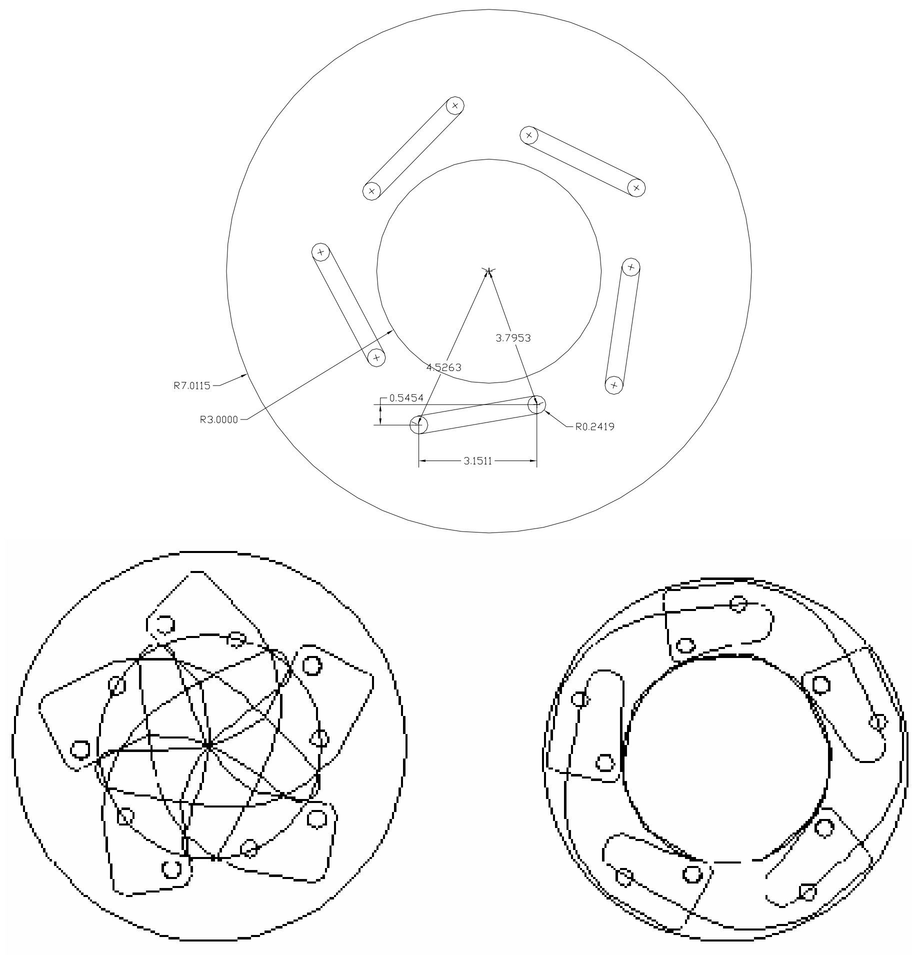

Figure 5. Design of 'IRIS'

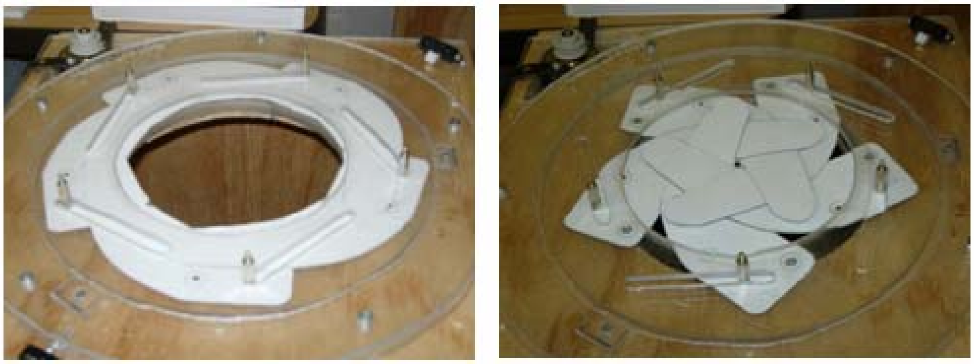

Figure 6. 'IRIS' as designed (open and close views)

SOFTWARE

Micro-IDE, a Windows-based coding environment provided by BiPOM, was used to develop the software for STAN. Both assembly code and the Micro-C language were used for programming.

The program was divided into the following modules:

- Routine for monitoring the methane level inside the trash can
- Routine for monitoring the weight of the trash can
- Routine for detecting presence at the trash can for opening the lid
- Timing routine for delaying the closure of the lid
- Motor control routine for opening the lid
- Motor control routine for reversing polarity of the motor and closing the lid
- Routine for scanning the keypad for user input
- Routine for monitoring status of the door contact
- Routine for monitoring service mode status
- User interface shell

The program was developed based on pseudo code logic provided below.

```
Interrupt routine for timer0
{
If ((door contact == open) & (service mode == normal))
{
      Pause program until Door contact is closed
}

Obtain distance sensor reading

If (distance < threshold)
```

```
{
      Open Trashcan lid
      Begin time delay
}
If ((time delay > 5 sec) & (distance > threshold))
{
      Close Trashcan lid
}
If(weight > threshold)
{
      Display Service Message on LCD
      Pause Program until unit serviced
}
If(methane > threshold)
{
      Display Service Message on LCD
      Pause Program until unit serviced
}

Main()
{
Initialize variables, etc.
Display “Enter PIN” on LCD
Scan Keypad for PIN
Check entry against coded PIN

If (wrong PIN)
{
      Display “Wrong PIN”
      Return to “Display Enter PIN”
}
If (correct pin)
{
      Set service mode = Service
      Display “Service mode, press # to exit” on LCD
      Wait for # from keypad
      Set service mode = Normal
}
```

The functions to achieve each system subtask were programmed using Micro C, a variant of C programming language used to program microcontrollers. The functions are mostly called from either the main() function or from an interrupt. The functions ScanKeypad(), OpenLid(), CloseLid(), and the control functions WriteCADC(), ReadCADC(), ReadADC(), Set10bitPWM() are used. The variables LidStatus, OpenCounter, ServiceMode are used to determine the various system states.

Function ScanKeypad()

```
char ScanKeypad()
{
    char row;
    char col;
```

```
    col = 0;                             // current column

    for( row=0; row<MAX_ROWS; row++ )    // monitor rows
    {
        P2 = RowTable[row];

        if( !(P2 & 0x80) )col = 4;       // monitor columns
        if( !(P2 & 0x40) )col = 3;
        if( !(P2 & 0x20) )col = 2;
        if( !(P2 & 0x10) )col = 1;
        if( col != 0 )
        {
            delay(500);
            return KeyTable[col-1 + row*MAX_COLS];    // return key value
        }
    }
    return 0;
}
```

The ScanKeypad() function returns the value that is pressed on the keypad depending on the row-column input from the keypad. This function is called from the interrupt service routine to check the PIN for validation to enter the maintenance mode.

Function Openlid():

```
/*Function: OpenLid - Turns motor on to open lid*/

void OpenLid()
{

      setbit(MOTORDIR);        //start motor in open direction
      clrbit(MOTORONOFF);
      while(LSOPEN & 0x04);    //monitor distance
      setbit(MOTORONOFF);      //stop motor
      LidStatus = 'O';
      return;
}
```

The function openlid() activates the motor in a particular direction so that the lid opens. The LidStatus is set to ‘O’ to indicate the lid is open.

Function Closelid():

```
/*Function: CloseLid - Turns motor on to close lid*/

void CloseLid()
{

      clrbit(MOTORDIR);        //start motor in close direction
      clrbit(MOTORONOFF);
      while(LSCLOSED & 0x08); //monitor distance
      setbit(MOTORONOFF);      //stop motor
      setbit(MOTORDIR);
      LidStatus = 'C';
```

```
      return;
}
```

This function does the opposite of Openlid() function. It sets the status of the lid to 'C' to indicate the lid is closed.

Interrupt module

```
INTERRUPT(_TF0_) TIMER0()
{
      register UBYTE cadc;
      register UINT adc;
      WriteCADC(0x80);

      cadc=0;
      ReadADC(ADC_UNIT+cadc,&adc);
      if(adc < 700)
      {
            LCD_Init();                   //Initialize LCD display (clear)
            LCD_WriteCtrl(0x02);          // Set Top Line
            LCD_printf("Trash Full");
            LCD_WriteCtrl(0xC0);          // Set Bottom Line
            LCD_printf ("Please Service");
            while(adc < 700)
            {
                  ReadADC(ADC_UNIT + cadc, &adc);
            }
            if(ServiceMode == 'S')
            {
                  LCD_Init();
                  LCD_WriteCtrl(0x02);
                  LCD_printf("Service Mode");         //write to LCD
                  LCD_WriteCtrl(0xC0);
                  LCD_printf("Press # to Exit");
             }
            if(ServiceMode == 'N')
            {
                  LCD_Init();                         // Initialize LCD
                  LCD_WriteCtrl(0x02);                // Set Top Line
                  LCD_printf( "Please enter PIN:");   // Prompt user for
                                                      // password
                  LCD_WriteCtrl(0xC0);                // Set bottom line
            }
      }

    cadc = 1;
    ReadADC(ADC_UNIT+cadc,&adc);
      if(adc > 100)
      {
            OpenLid();
            OpenCounter = 1;
            LCD_Init();                   //initialize LCD display (clear)
      }
      if((LidStatus == 'O') & (OpenCounter > 0))
      {
```

```
            OpenCounter++;
      }
      if((LidStatus == 'O') & (OpenCounter > 50))
      {
            OpenCounter = 0;
            CloseLid();
      }


    cadc = 4;
    ReadADC(ADC_UNIT+cadc,&adc);
    if(adc > 100)
    {
            LCD_Init();                   // Initialize LCD display (clear)
            LCD_WriteCtrl(0x02);          // Set Top Line
            LCD_printf("Methane Detected");
            LCD_WriteCtrl(0xC0);          // Set Bottom Line
            LCD_printf ("Please Service");
            while(adc > 90)
            {
                  ReadADC(ADC_UNIT+cadc,&adc);
            }
            if(ServiceMode == 'S')
            {
                  LCD_Init();
                  LCD_WriteCtrl(0x02);
                  LCD_printf("Service Mode");
                  LCD_WriteCtrl(0xC0);
                  LCD_printf("Press # to Exit");
        }
        if(ServiceMode == 'N')
        {
            LCD_Init();                         // Initialize LCD
            LCD_WriteCtrl(0x02);                // Set Top Line
            LCD_printf( "Please enter PIN:");   // Prompt user for
                                                // password
            LCD_WriteCtrl(0xC0);                // Set bottom line
        }
      }

      if (ServiceMode == 'N')
      {
            if(DOORCONTACT & 0x10)
            {
                  LCD_Init();
                  LCD_WriteCtrl(0x02);          // Set Top Line
                  LCD_printf("Please Close");
                  LCD_WriteCtrl(0xC0);          // Set Bottom Line
                  LCD_printf ("Service Door");
                  while (DOORCONTACT & 0x10);
                  LCD_Init();                         //Initialize LCD
                  LCD_WriteCtrl(0x02);                //Set Top Line
                  LCD_printf( "Please enter PIN:");//Prompt user for password
                  LCD_WriteCtrl(0xC0);                //Set bottom line
          }
       }
```

```
      /*Reinitialize Timer 0 to Value 0 (Max Time between Samples)*/

      TH0 = 0x00;
      TL0 = 0x00;
}
```

The timer interrupt module keeps reading the various input values from the ADC to which the sensors are connected and invoke the necessary functions if a pre-wired condition is met.

Function main():

```
void main()
{
      register UBYTE cadc;
      register UINT  adc;
      int var1;
      char key[15];                       //array for keypad entries
      int charcount;                      //for tracking entry length
      int k;                              //misc counter variable
      int j;                              //misc counter variable

      ServiceMode = 'N';
      var1 = 5;

      /* Initialize UART to 19200 baud rate */

      serinit(19200>>1);

      I2C_Repair();
      #ifndef EXT_REF
            WriteCADC(0xF0);        // VREF = 5V
                                    // An0, An1, An2, An3, An4
                                    // Please deinstall J8 and J9 jumpers
      #else
            WriteCADC(0xF8);        // VREF = 4.095V
                                    // An0, An1, An2 = GND, An3 = VREF, An4
                                    // Please install J8 and J9 jumpers
      #endif
      Set10bitPWM (0);              // Vee = 2.5V, pin#3 of J3
      ReadCADC(&cadc);
      printf ("\nCADC = %02x",cadc);
      LCD_Init();                   // Initialize LCD
      OpenLid();

      TH0 = 0x00;                   //Initialize Counter0 to value 0
      TL0 = 0x00;
      TMOD = TMOD & 0xF0;           //Clear Timer0 settings while
                                    //preserving Timer1
      TMOD = TMOD | 0x01;           //Set Timer0 settings while
                                    //preserving Timer1
      TCON = TCON | 0x10;           //Start Timer0 without
                                    //modifying other settings
      IE = 0x82;                    //Enable the Interrupt for Timer0
```

```
for(;;)
{
      LCD_Init();                          // Initialize LCD
      LCD_WriteCtrl(0x02);                 // Set Top Line
      LCD_printf( "Please enter PIN:");    //Prompt user for
                                     //password
      charcount = 0;                 //initialize character count
                                     //variable
      LCD_WriteCtrl(0xC0);           // Set bottom line

// scanning the 4X4 matrix keypad. Displays the key that was
//pressed

for( ;; )                            //Begin scanning keypad
{
    key[charcount] = ScanKeypad();   //Start Scanning Keypad for
                                     //values

    if( key[charcount] )
    {
        if ( key[charcount] == '#') //Check for Escape Character
            break;
        // Display the key that was pressed on the LCD screen
        LCD_printf( "%c", key[charcount] );
        charcount++;    //Update charcount for pin verification
      }

}

j = 0;                  //Initialize j for checking PIN length

for(k = 0; k < charcount; k++)//Start Processing Entered PIN
{
    if(key[k] == PIN[k])//check if pin array matches key array
    {
          j++;    //if matched, increment j for pin length verification

            //if pin matches and is correct length, proceed, else fail.

            if((j==charcount)&(j==4))
            {
              LCD_Init();     //initialize LCD display (clear)
              LCD_WriteCtrl(0xC0);        // Set Bottom Line
              LCD_printf ("PIN accepted");//Display
                                          //verification message
              delay(1200);                //wait
              ServiceMode = 'S';
              LCD_Init();
              LCD_WriteCtrl(0x02);
              LCD_printf("Service Mode");
              LCD_WriteCtrl(0xC0);
              LCD_printf("Press # to Exit");

              for(;;)                //Begin scan keypad
              {
                  if(ScanKeypad() == '#')
```

```
                        break;

                //if # pressed, escape to beginning (pin check)
                  }
                  ServiceMode = 'N';
               }
            }

            else                    //Pin Failure Routine
            {
                  LCD_Init();
            //Clear the Display, display incorrect pin, return to pin entry
                  LCD_WriteCtrl(0xC0);
                  LCD_printf ("Incorrect PIN");
                  delay(1200);
                  break;
            }
      }

    }

}
```

This is the main function where the timer mode, etc. settings are made. This is the function where the PIN is initially checked.

The functional flowchart is shown in Figure 7.

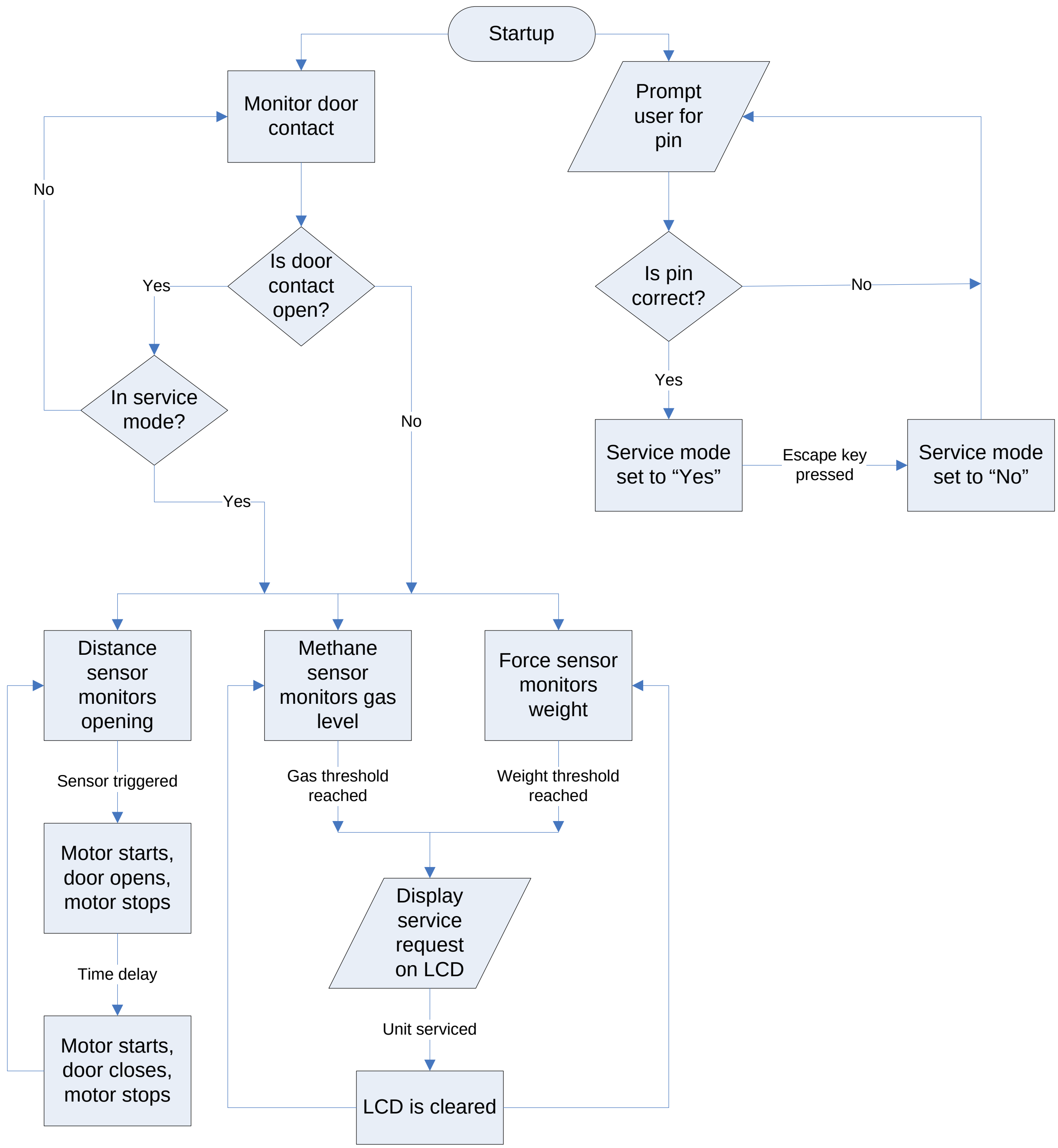

Figure 7. Software functional flowchart

FEATURES AND SPECIFICATIONS OF STAN

The features of STAN are as follows:

- Touchless opening is triggered by a distance sensor
- Methane detector which alerts in the case of decomposing material
- Force sensor which alerts if maximum weight capacity has been reached
- LCD display and keypad that maintenance can change from normal to service mode
- Opening time is less than 3 seconds from a fully closed state
- Assuming the unit would operate on a 38AH battery when used in remote locations, the unit could operate for 5 days on a single battery assuming activation once every 30 minutes
- Plywood was used for manufacturing the outer shroud.
- Sheet metal was used for the ‘iris’ blades, as they are needed to be both strong and thin
- Plexiglas® was used for the ‘iris’ base ring and actuating ring for its strength and ease of forming
- Velcro was used to mount the hardware components inside the enclosure for easy removal and servicing as shown in Figure 8.

Figure 8. A fully functional STAN

The specifications are as follows:

- The motor to drive the ‘iris’ mechanism is specified to have 60-120 RPM at 12 volts DC.
- The motor is rated for use up to 38 volts DC.
- The maximum current draw of the system was 1 A.
- Normal operating current was less than 300 mA.

- The assumed duty cycle of high power operation is less than 5%.
- The shroud was constructed out of standard plywood and had the dimensions of 18 inches square on top, and 24 inches high.

FUTURE WORK

Additional work is focused on enhancements to create a compressing mechanism to minimize volume, and to develop a network of smart trash cans in a specified zone, controlled by a single operator.

Future work on this project is conceived to include an anti-terror module. The rise of terrorism in the world would make this addition very realistic in the market today. The anti-terror module would be a bomb detecting sensor to detect the presence of explosives in the trash can and send alerting signals to local authorities. While this design would not prevent the explosion, it would allow for a faster response by authorities. This could result in the possible capture of the assailant. Use of the solar power to prolong the life of rechargeable battery is another consideration.

CONCLUSION

STAN project was a learning experience for the team. STAN prototype was fully functional and is expected to be a good prospective commercial product.

REFERENCES

[1]. BiPOM Electronics, Inc. Mini-Max/51-C2 Technical Manual
http://www.bipom.com/documents/boards/minimax51c2.pdf [Accessed October 10, 2006].
[2]. Sharp GP2D12 IR Sensor Specifications
http://www.lynxmotion.com/ [Accessed October 10, 2006].
[3]. FlexiForce® Force Sensor Specifications
http://www.tekscan.com/flexiforce/flexiforce.html [Accessed October 10, 2006].
[4]. Figaro NGM2611 Specifications
http://www.figaro.co.jp/en/pdf/NGM2611ProductInfo1205.pdf [Accessed October 10, 2006].