Tele-Operated Alarm Monitoring Interface

By
Paul Pankratz

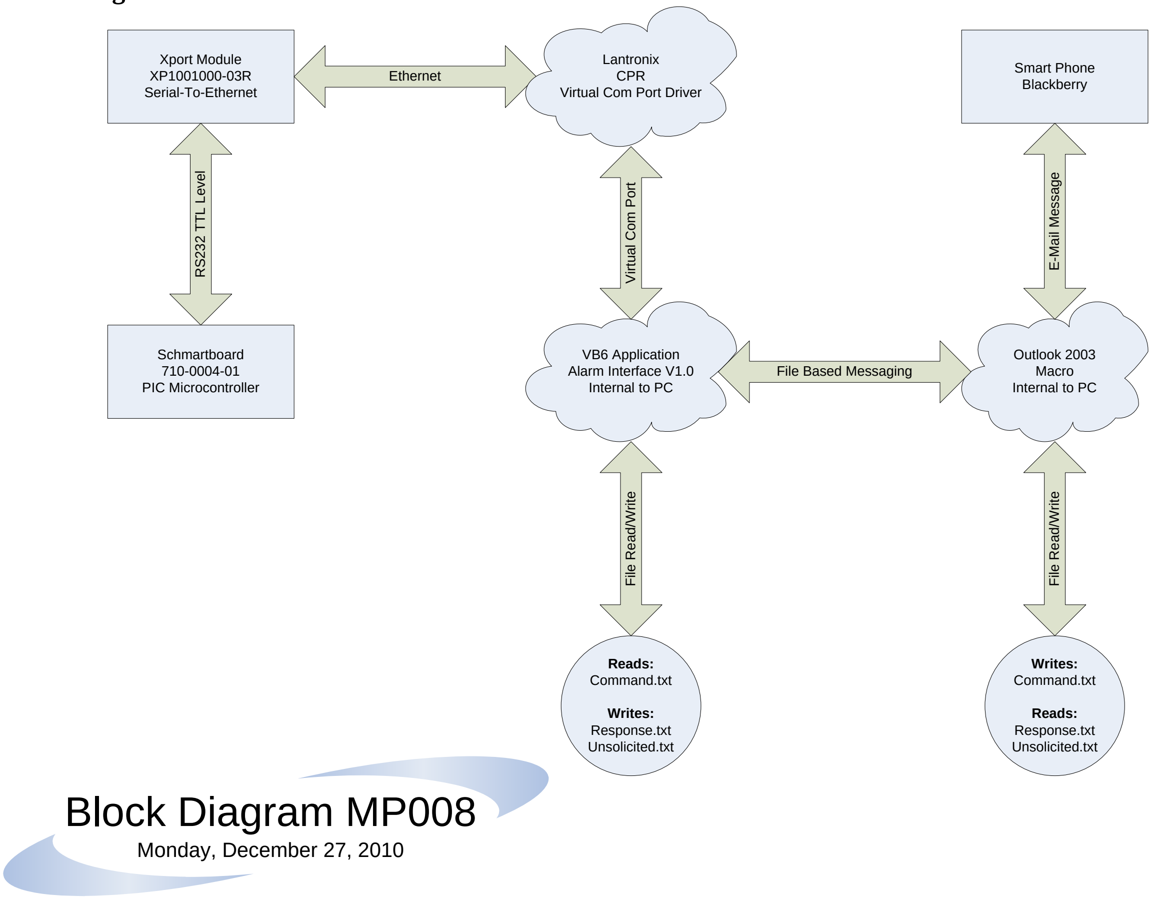
Project Number:
MP008

Project Description:
Quite awhile ago I decided to start trimming the fat from our household budget and looking for ways to reduce or eliminate recurring daily costs as it pertains to day to day life.

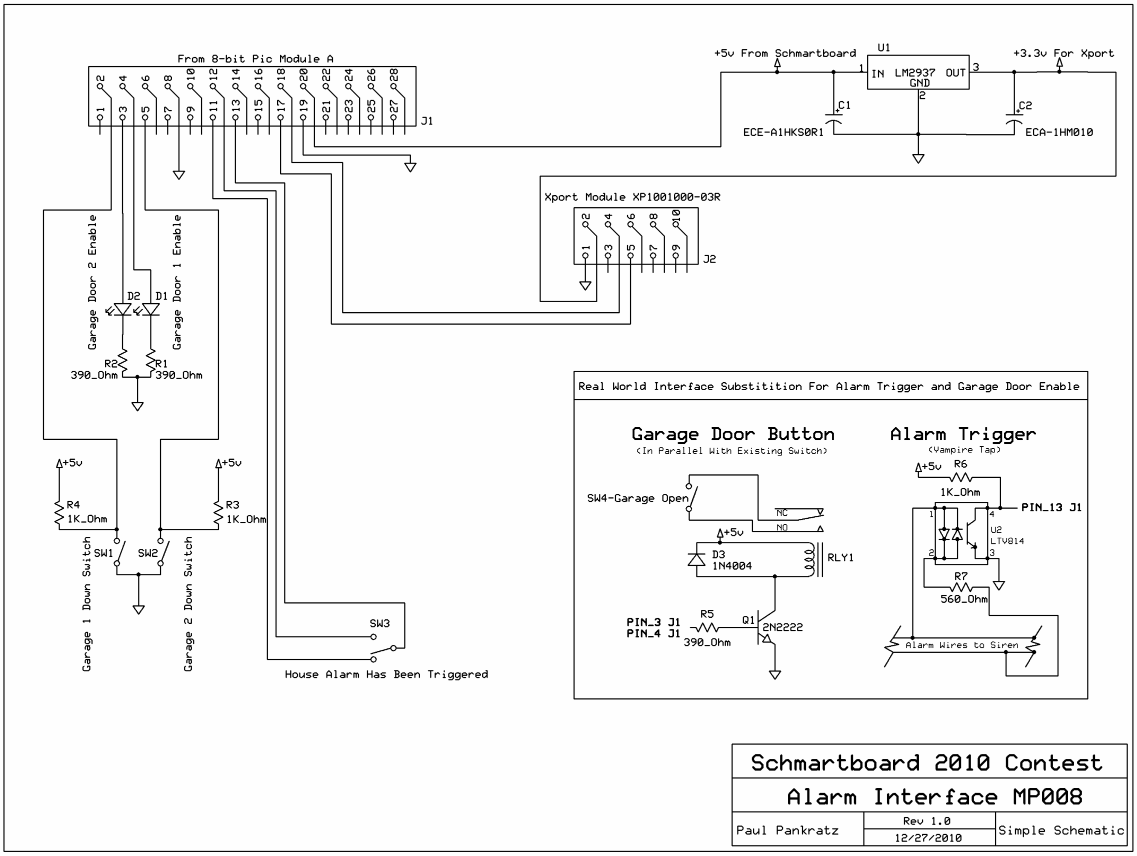
My long term goal is to live off the grid, but that requires quite an investment so I started smaller. The first thing to go was the costly telephone bill which after several months of research I came to the conclusion that Oomah was the best choice a little up front cost then $12/year there after. My second project led me to this contest entry which was eliminating our alarm monitoring service. I thought to myself, why am I paying someone $30/month to call me if my house alarm system goes off? I am a systems engineer who specializes in embedded systems by profession so this project was right up my alley.

I turned to Schmartboard for my prototype, what more could a guy need to get started. The thought was that I would use the Schmartboard for the low level bit twiddling and interfacing to the real world. I would use an Xport serial to Ethernet device to handle the communication to and from the computer. I would then use VB6 as my front end to handle the communications bit and message dispatching. And lastly I would use Outlook 2003 to run a macro that would handle the email to VB6 application interface. Sounds pretty confusing right? Below is a high level block diagram of what I tried to describe above:

Block Diagram:



Schematic:



Bill of Materials:

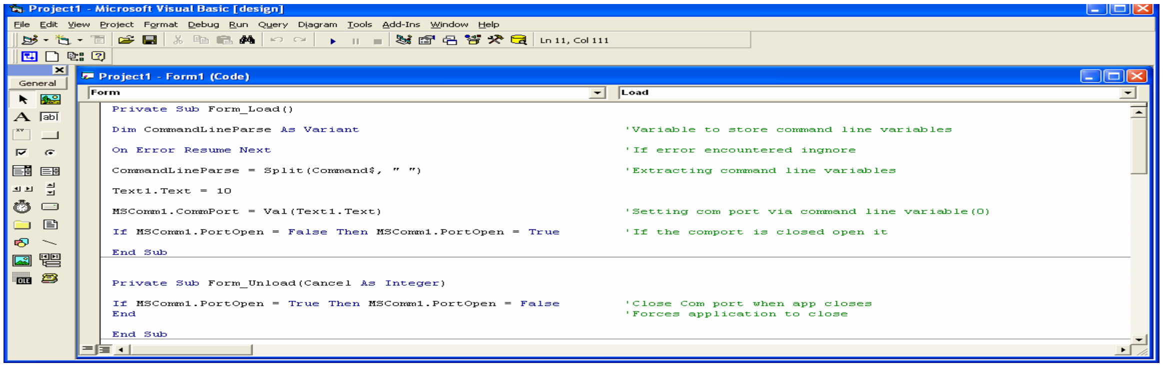
C1	ECE-A1HKS0R1	P984-ND
C2	ECA-1HM010	P5174-ND
D1	Garage Door 1 Enable	P373-ND
D2	Garage Door 2 Enable	P373-ND
D3	1N4004	1N4004DITR-ND
J1	From 8-bit Pic Module A	710-0004-01
J2	Xport Module XP1001000-03R	515-XP1001000-03R
Q1	2N2222	497-2598-ND
R1	390_Ohm	390EBK-ND
R2	390_Ohm	390EBK-ND
R3	1K_Ohm	CF18JT1K00TR-ND
R4	1K_Ohm	CF18JT1K00TR-ND
R5	390_Ohm	390EBK-ND
R6	1K_Ohm	CF18JT1K00TR-ND
R7	560_Ohm	P560CACT-ND
RLY1	5V Relay SPDT	Z1014-ND
SW1	Garage 1 Down Switch	SW156-ND
SW2	Garage 2 Down Switch	SW156-ND
SW3	House Alarm Has Been Triggered	SW156-ND
SW4-Garage Open	Garage Door Button	Existing in Garage
U1	LM2937	LM2937ET-3.3-ND
U2	LTV814	160-1344-5-ND

Source Code Explanation:

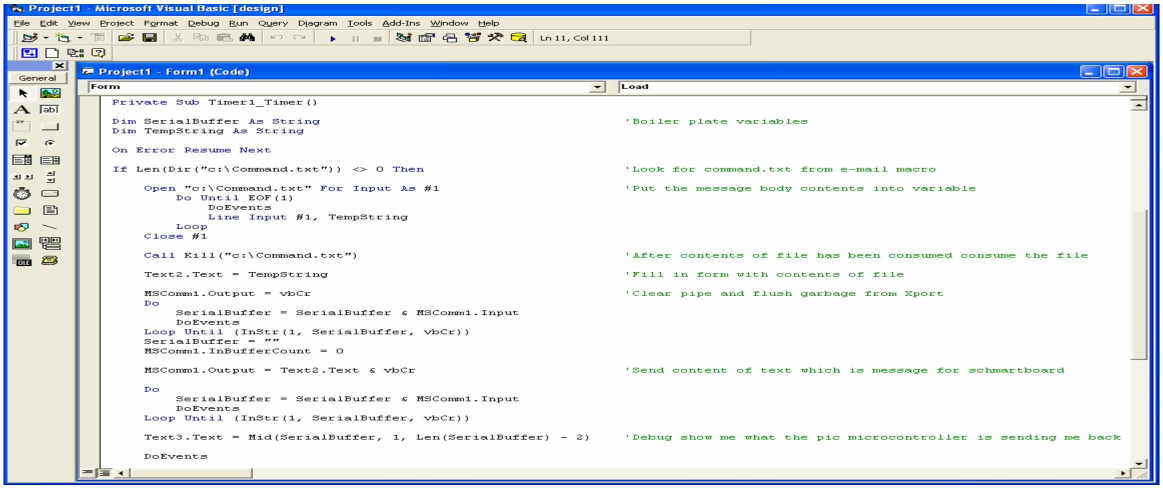
The source codes included in this document are screen shots. Your best bet for seeing them as they were intended would be to use the proper IDE and view the file attachments as follows:

VB6 = Microsoft visual studio.
VBA = Outlook 2003 macro editor.
PIC = MPLAB IDE V8.60 running CCS PCWHD C Compiler v4.093.
IETIMER = Needed for a VBA user form timer can be freely downloaded search Google.
CPR Manager = Virtual com port used by Xport can be freely downloaded search Google.

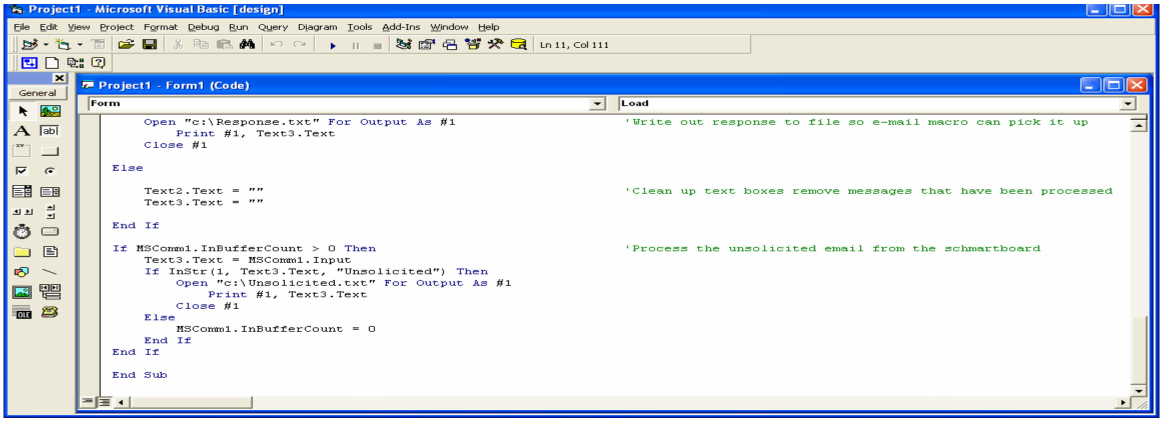
Source Code VB6:


Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 11, Col 111
General
Project1 - Form1 (Code)
Form
Load

```
Private Sub Form_Load()

Dim CommandLineParse As Variant                                    'Variable to store command line variables

On Error Resume Next                                               'If error encountered ingnore

CommandLineParse = Split(Command$, " ")                            'Extracting command line variables

Text1.Text = 10

MSComm1.CommPort = Val(Text1.Text)                                 'Setting com port via command line variable(0)

If MSComm1.PortOpen = False Then MSComm1.PortOpen = True           'If the comport is closed open it

End Sub

Private Sub Form_Unload(Cancel As Integer)

If MSComm1.PortOpen = True Then MSComm1.PortOpen = False           'Close Com port when app closes
End                                                                'Forces application to close

End Sub
```

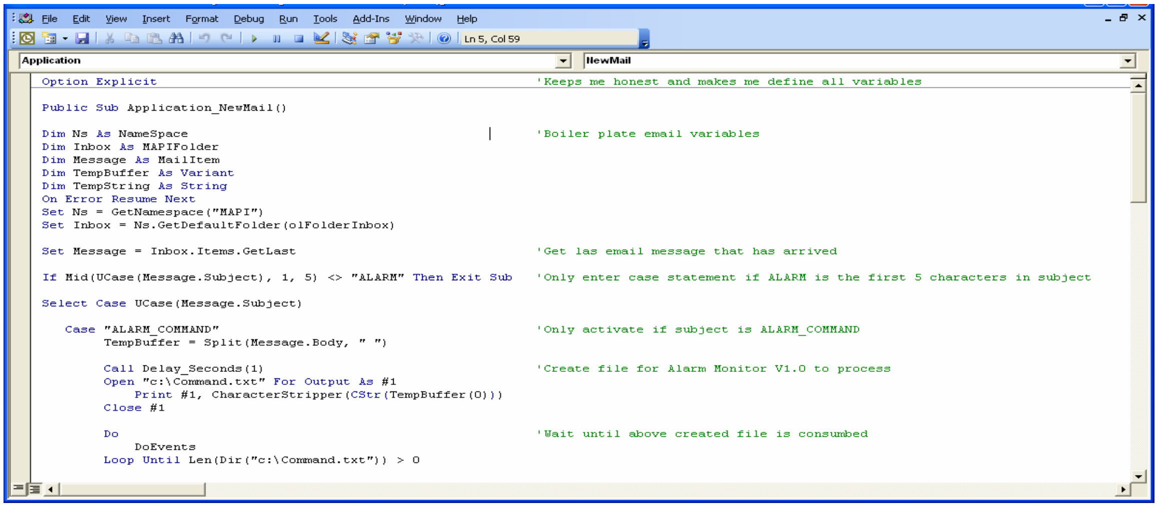

Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 11, Col 111
General
Project1 - Form1 (Code)
Form
Load

```
Private Sub Timer1_Timer()

Dim SerialBuffer As String                                         'Boiler plate variables
Dim TempString As String

On Error Resume Next

If Len(Dir("c:\Command.txt")) <> 0 Then                            'Look for command.txt from e-mail macro

    Open "c:\Command.txt" For Input As #1                          'Put the message body contents into variable
        Do Until EOF(1)
            DoEvents
            Line Input #1, TempString
        Loop
    Close #1

    Call Kill("c:\Command.txt")                                    'After contents of file has been consumed consume the file

    Text2.Text = TempString                                        'Fill in form with contents of file

    MSComm1.Output = vbCr                                          'Clear pipe and flush garbage from Xport
    Do
        SerialBuffer = SerialBuffer & MSComm1.Input
        DoEvents
    Loop Until (InStr(1, SerialBuffer, vbCr))
    SerialBuffer = ""
    MSComm1.InBufferCount = 0

    MSComm1.Output = Text2.Text & vbCr                             'Send content of text which is message for schmartboard

    Do
        SerialBuffer = SerialBuffer & MSComm1.Input
        DoEvents
    Loop Until (InStr(1, SerialBuffer, vbCr))

    Text3.Text = Mid(SerialBuffer, 1, Len(SerialBuffer) - 2)       'Debug show me what the pic microcontroller is sending me back

    DoEvents
```

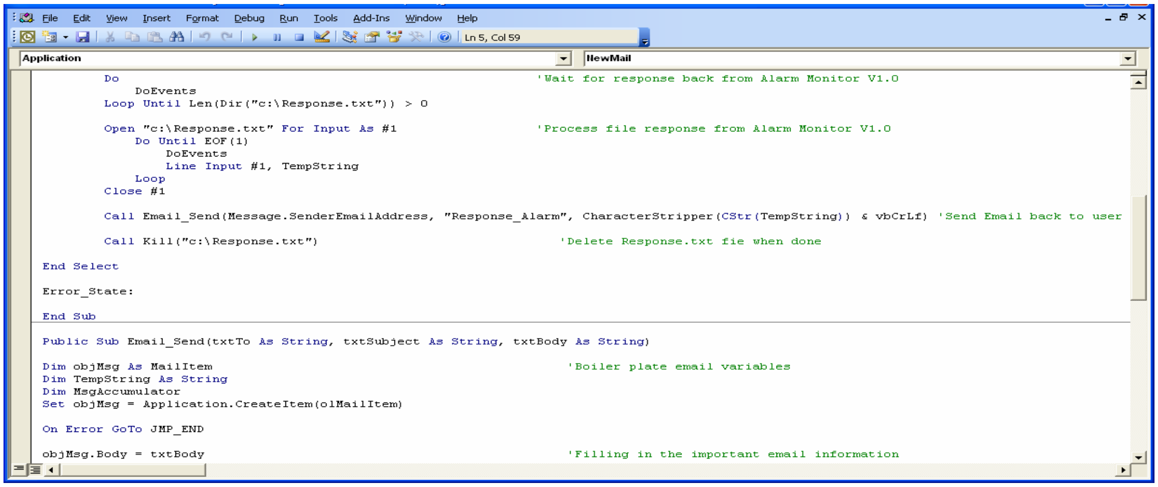

Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 11, Col 111
General
Project1 - Form1 (Code)
Form
Load

```
    Open "c:\Response.txt" For Output As #1                        'Write out response to file so e-mail macro can pick it up
        Print #1, Text3.Text
    Close #1

Else

    Text2.Text = ""                                                'Clean up text boxes remove messages that have been processed
    Text3.Text = ""

End If

If MSComm1.InBufferCount > 0 Then                                  'Process the unsolicited email from the schmartboard
    Text3.Text = MSComm1.Input
    If InStr(1, Text3.Text, "Unsolicited") Then
        Open "c:\Unsolicited.txt" For Output As #1
            Print #1, Text3.Text
        Close #1
    Else
        MSComm1.InBufferCount = 0
    End If
End If

End Sub
```

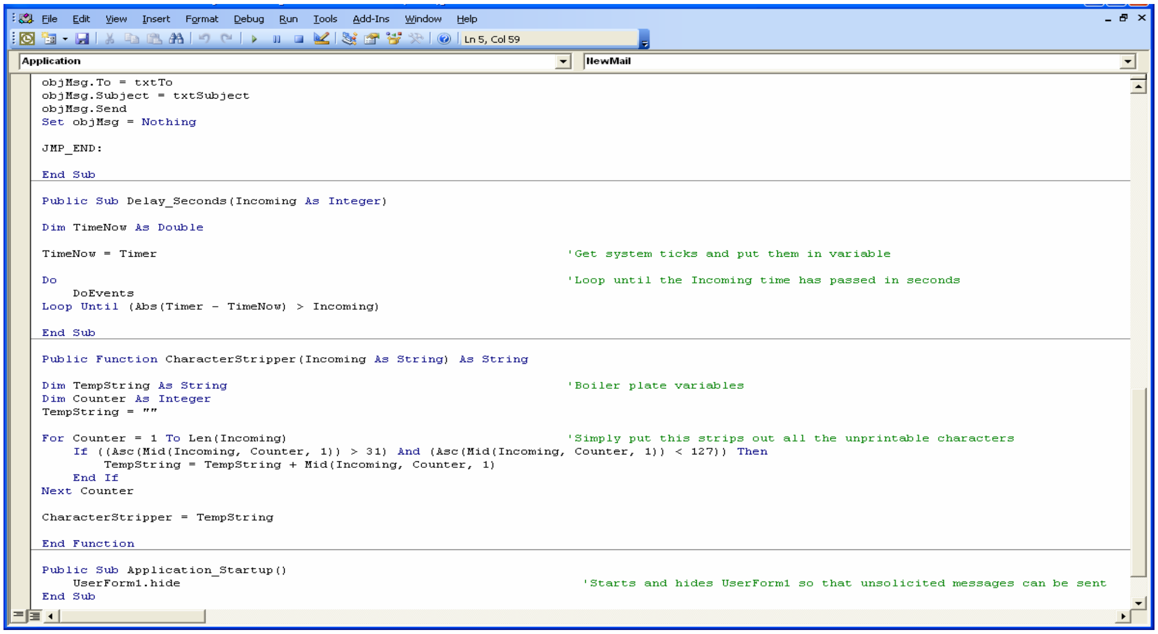
Source Code Email VBA:



```
Option Explicit                                                 'Keeps me honest and makes me define all variables

Public Sub Application_NewMail()

Dim Ns As NameSpace                                             'Boiler plate email variables
Dim Inbox As MAPIFolder
Dim Message As MailItem
Dim TempBuffer As Variant
Dim TempString As String
On Error Resume Next
Set Ns = GetNamespace("MAPI")
Set Inbox = Ns.GetDefaultFolder(olFolderInbox)

Set Message = Inbox.Items.GetLast                               'Get las email message that has arrived

If Mid(UCase(Message.Subject), 1, 5) <> "ALARM" Then Exit Sub   'Only enter case statement if ALARM is the first 5 characters in subject

Select Case UCase(Message.Subject)

   Case "ALARM_COMMAND"                                         'Only activate if subject is ALARM_COMMAND
        TempBuffer = Split(Message.Body, " ")

        Call Delay_Seconds(1)                                   'Create file for Alarm Monitor V1.0 to process
        Open "c:\Command.txt" For Output As #1
            Print #1, CharacterStripper(CStr(TempBuffer(0)))
        Close #1

        Do                                                      'Wait until above created file is consumbed
            DoEvents
        Loop Until Len(Dir("c:\Command.txt")) > 0
```

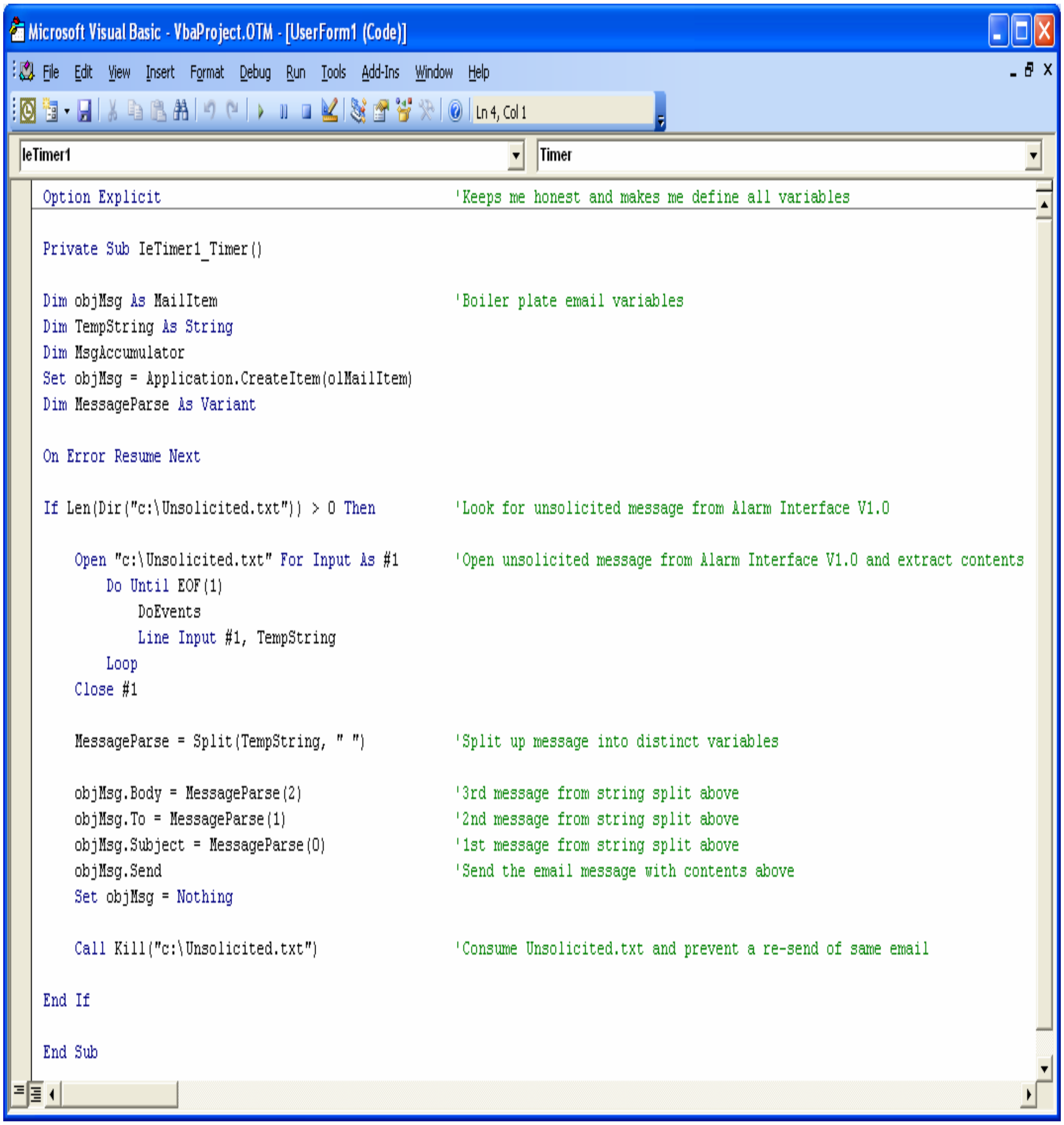


```
        Do                                                      'Wait for response back from Alarm Monitor V1.0
            DoEvents
        Loop Until Len(Dir("c:\Response.txt")) > 0

        Open "c:\Response.txt" For Input As #1                  'Process file response from Alarm Monitor V1.0
            Do Until EOF(1)
                DoEvents
                Line Input #1, TempString
            Loop
        Close #1

        Call Email_Send(Message.SenderEmailAddress, "Response_Alarm", CharacterStripper(CStr(TempString)) & vbCrLf) 'Send Email back to user

        Call Kill("c:\Response.txt")                               'Delete Response.txt fie when done

End Select

Error_State:

End Sub

Public Sub Email_Send(txtTo As String, txtSubject As String, txtBody As String)

Dim objMsg As MailItem                                              'Boiler plate email variables
Dim TempString As String
Dim MsgAccumulator
Set objMsg = Application.CreateItem(olMailItem)

On Error GoTo JMP_END

objMsg.Body = txtBody                                               'Filling in the important email information
```



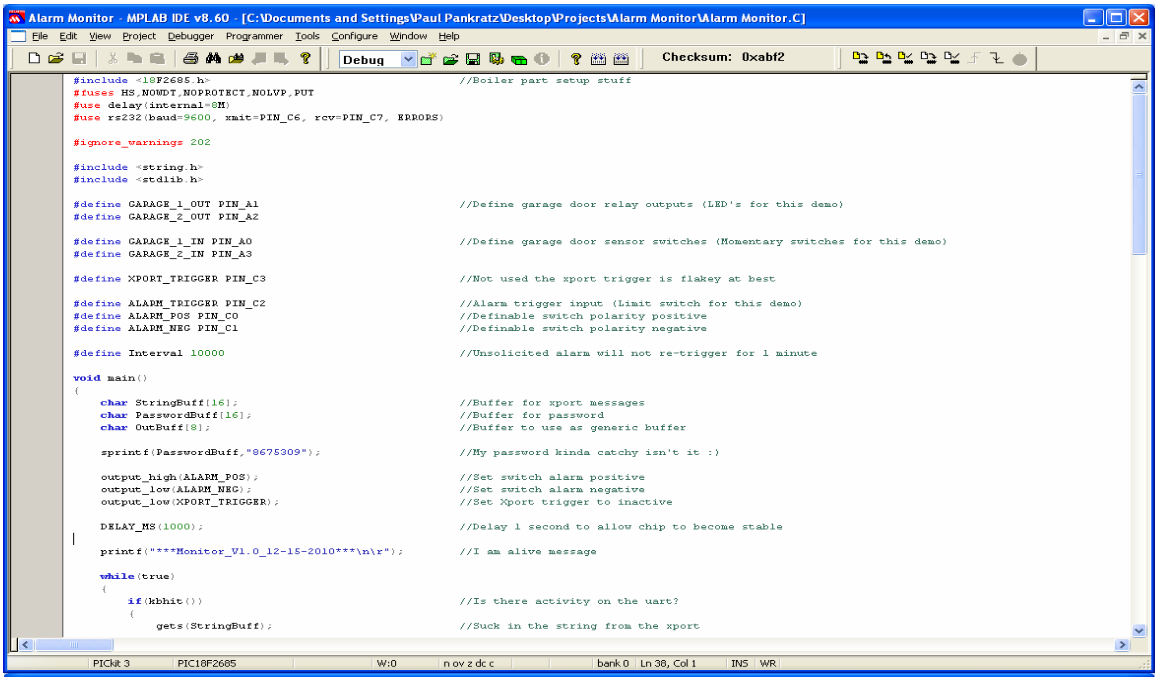
```
objMsg.To = txtTo
objMsg.Subject = txtSubject
objMsg.Send
Set objMsg = Nothing

JMP_END:

End Sub

Public Sub Delay_Seconds(Incoming As Integer)

Dim TimeNow As Double

TimeNow = Timer                                                     'Get system ticks and put them in variable

Do                                                                  'Loop until the Incoming time has passed in seconds
    DoEvents
Loop Until (Abs(Timer - TimeNow) > Incoming)

End Sub

Public Function CharacterStripper(Incoming As String) As String

Dim TempString As String                                            'Boiler plate variables
Dim Counter As Integer
TempString = ""

For Counter = 1 To Len(Incoming)                                    'Simply put this strips out all the unprintable characters
    If ((Asc(Mid(Incoming, Counter, 1)) > 31) And (Asc(Mid(Incoming, Counter, 1)) < 127)) Then
        TempString = TempString + Mid(Incoming, Counter, 1)
    End If
Next Counter

CharacterStripper = TempString

End Function

Public Sub Application_Startup()
    UserForm1.hide                                                    'Starts and hides UserForm1 so that unsolicited messages can be sent
End Sub
```

Source Code Email VBA User Form:

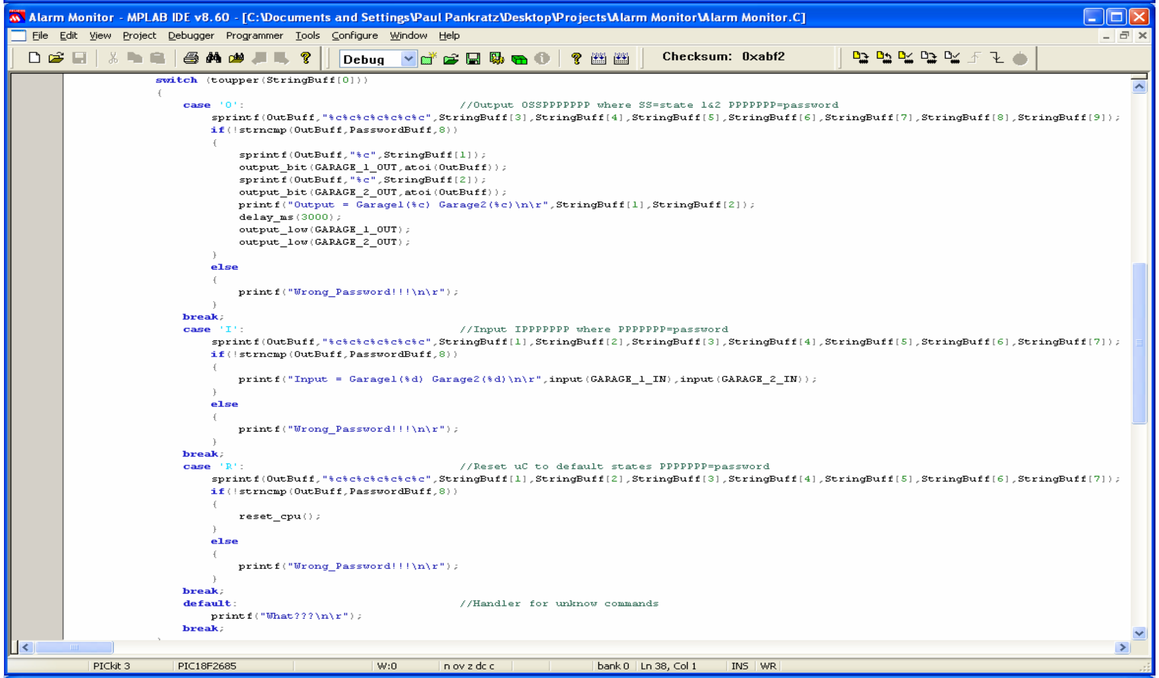

Microsoft Visual Basic - VbaProject.OTM - [UserForm1 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 4, Col 1
IeTimer1
Timer

```
Option Explicit                                     'Keeps me honest and makes me define all variables

Private Sub IeTimer1_Timer()

Dim objMsg As MailItem                              'Boiler plate email variables
Dim TempString As String
Dim MsgAccumulator
Set objMsg = Application.CreateItem(olMailItem)
Dim MessageParse As Variant

On Error Resume Next

If Len(Dir("c:\Unsolicited.txt")) > 0 Then          'Look for unsolicited message from Alarm Interface V1.0

    Open "c:\Unsolicited.txt" For Input As #1       'Open unsolicited message from Alarm Interface V1.0 and extract contents
        Do Until EOF(1)
            DoEvents
            Line Input #1, TempString
        Loop
    Close #1

    MessageParse = Split(TempString, " ")           'Split up message into distinct variables

    objMsg.Body = MessageParse(2)                   '3rd message from string split above
    objMsg.To = MessageParse(1)                     '2nd message from string split above
    objMsg.Subject = MessageParse(0)                '1st message from string split above
    objMsg.Send                                     'Send the email message with contents above
    Set objMsg = Nothing

    Call Kill("c:\Unsolicited.txt")                 'Consume Unsolicited.txt and prevent a re-send of same email

End If

End Sub
```

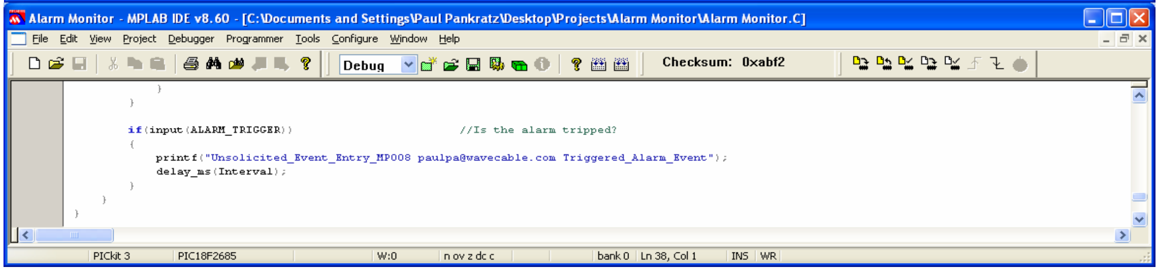
Source Code PIC Microcontroller:


Alarm Monitor - MPLAB IDE v8.60 - [C:\Documents and Settings\Paul Pankratz\Desktop\Projects\Alarm Monitor\Alarm Monitor.C]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xabf2
PICkit 3 PIC18F2685 W:0 n ov z dc c bank 0 Ln 38, Col 1 INS WR

```c
#include <18F2685.h>                                    //Boiler part setup stuff
#fuses HS,NOWDT,NOPROTECT,NOLVP,PUT
#use delay(internal=8M)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, ERRORS)

#ignore_warnings 202

#include <string.h>
#include <stdlib.h>

#define GARAGE_1_OUT PIN_A1                             //Define garage door relay outputs (LED's for this demo)
#define GARAGE_2_OUT PIN_A2

#define GARAGE_1_IN PIN_A0                              //Define garage door sensor switches (Momentary switches for this demo)
#define GARAGE_2_IN PIN_A3

#define XPORT_TRIGGER PIN_C3                            //Not used the xport trigger is flakey at best

#define ALARM_TRIGGER PIN_C2                            //Alarm trigger input (Limit switch for this demo)
#define ALARM_POS PIN_C0                                //Definable switch polarity positive
#define ALARM_NEG PIN_C1                                //Definable switch polarity negative

#define Interval 10000                                  //Unsolicited alarm will not re-trigger for 1 minute

void main()
{
    char StringBuff[16];                                //Buffer for xport messages
    char PasswordBuff[16];                              //Buffer for password
    char OutBuff[8];                                    //Buffer to use as generic buffer

    sprintf(PasswordBuff,"8675309");                    //My password kinda catchy isn't it :)

    output_high(ALARM_POS);                             //Set switch alarm positive
    output_low(ALARM_NEG);                              //Set switch alarm negative
    output_low(XPORT_TRIGGER);                          //Set Xport trigger to inactive

    DELAY_MS(1000);                                     //Delay 1 second to allow chip to become stable

    printf("***Monitor_V1.0_12-15-2010***\n\r");        //I am alive message

    while(true)
    {
        if(kbhit())                                     //Is there activity on the uart?
        {
            gets(StringBuff);                           //Suck in the string from the xport
```


Alarm Monitor - MPLAB IDE v8.60 - [C:\Documents and Settings\Paul Pankratz\Desktop\Projects\Alarm Monitor\Alarm Monitor.C]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xabf2
PICkit 3 PIC18F2685 W:0 n ov z dc c bank 0 Ln 38, Col 1 INS WR

```c
            switch (toupper(StringBuff[0]))
            {
                case 'O':                               //Output OSSPPPPPPP where SS=state 1&2 PPPPPPP=password
                    sprintf(OutBuff,"%c%c%c%c%c%c%c",StringBuff[3],StringBuff[4],StringBuff[5],StringBuff[6],StringBuff[7],StringBuff[8],StringBuff[9]);
                    if(!strncmp(OutBuff,PasswordBuff,8))
                    {
                        sprintf(OutBuff,"%c",StringBuff[1]);
                        output_bit(GARAGE_1_OUT,atoi(OutBuff));
                        sprintf(OutBuff,"%c",StringBuff[2]);
                        output_bit(GARAGE_2_OUT,atoi(OutBuff));
                        printf("Output = Garage1(%c) Garage2(%c)\n\r",StringBuff[1],StringBuff[2]);
                        delay_ms(3000);
                        output_low(GARAGE_1_OUT);
                        output_low(GARAGE_2_OUT);
                    }
                    else
                    {
                        printf("Wrong_Password!!!\n\r");
                    }
                break;
                case 'I':                               //Input IPPPPPPP where PPPPPPP=password
                    sprintf(OutBuff,"%c%c%c%c%c%c%c",StringBuff[1],StringBuff[2],StringBuff[3],StringBuff[4],StringBuff[5],StringBuff[6],StringBuff[7]);
                    if(!strncmp(OutBuff,PasswordBuff,8))
                    {
                        printf("Input = Garage1(%d) Garage2(%d)\n\r",input(GARAGE_1_IN),input(GARAGE_2_IN));
                    }
                    else
                    {
                        printf("Wrong_Password!!!\n\r");
                    }
                break;
                case 'R':                               //Reset uC to default states PPPPPPP=password
                    sprintf(OutBuff,"%c%c%c%c%c%c%c",StringBuff[1],StringBuff[2],StringBuff[3],StringBuff[4],StringBuff[5],StringBuff[6],StringBuff[7]);
                    if(!strncmp(OutBuff,PasswordBuff,8))
                    {
                        reset_cpu();
                    }
                    else
                    {
                        printf("Wrong_Password!!!\n\r");
                    }
                break;
                default:                                //Handler for unknow commands
                    printf("What???\n\r");
                break;
```

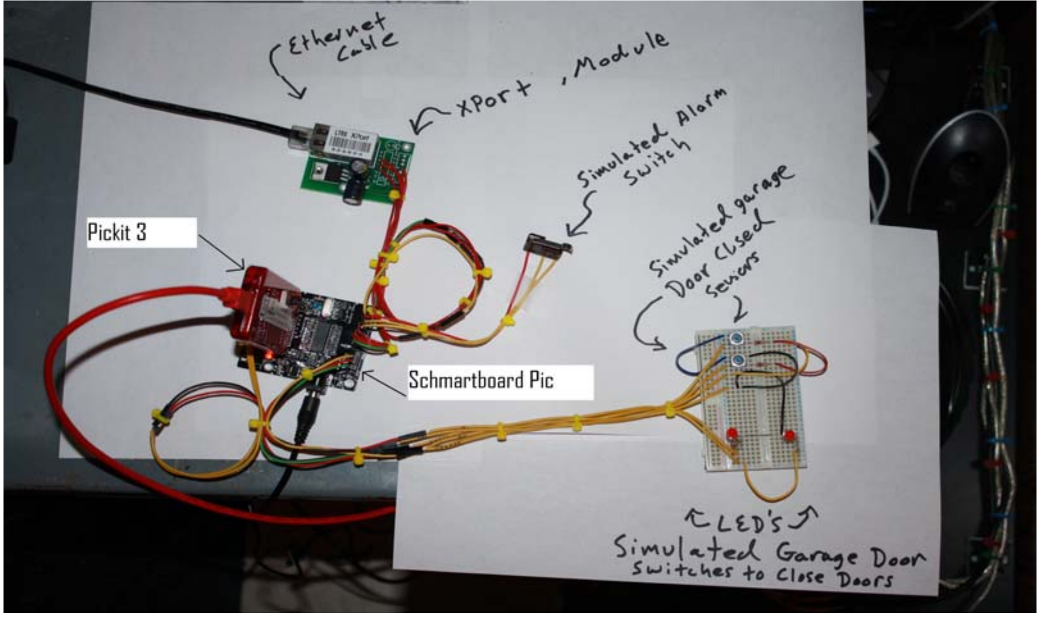

Alarm Monitor - MPLAB IDE v8.60 - [C:\Documents and Settings\Paul Pankratz\Desktop\Projects\Alarm Monitor\Alarm Monitor.C]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug
Checksum: 0xabf2
PICkit 3 PIC18F2685 W:0 n ov z dc c bank 0 Ln 38, Col 1 INS WR

```c
            }
        }

        if(input(ALARM_TRIGGER))                        //Is the alarm tripped?
        {
            printf("Unsolicited_Event_Entry_MP008 paulpa@wavecable.com Triggered_Alarm_Event");
            delay_ms(Interval);
        }
    }
}
```

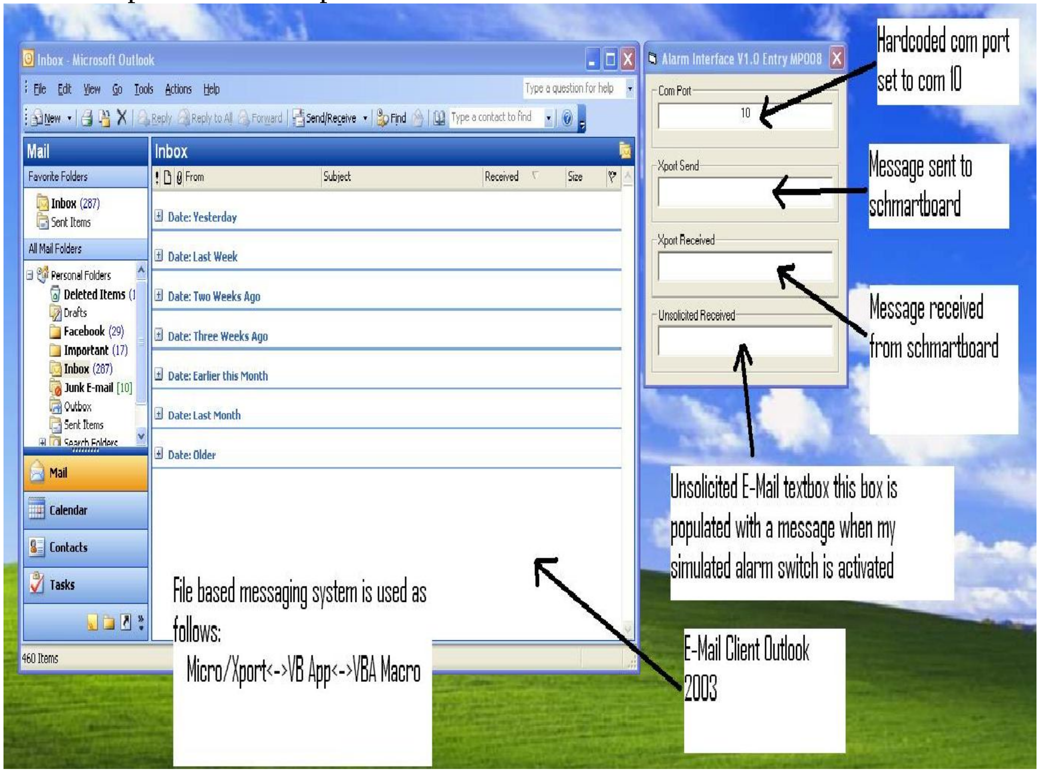
.

Pictures:

Project in prototype form on my Desk:


Ethernet Cable
XPort , Module
Simulated Alarm Switch
Pickit 3
Simulated garage Door Closed Sensors
Schmartboard Pic
LED's
Simulated Garage Door Switches to Close Doors

Screen capture of desktop:



Hardcoded com port set to com 10

Message sent to schmartboard

Message received from schmartboard

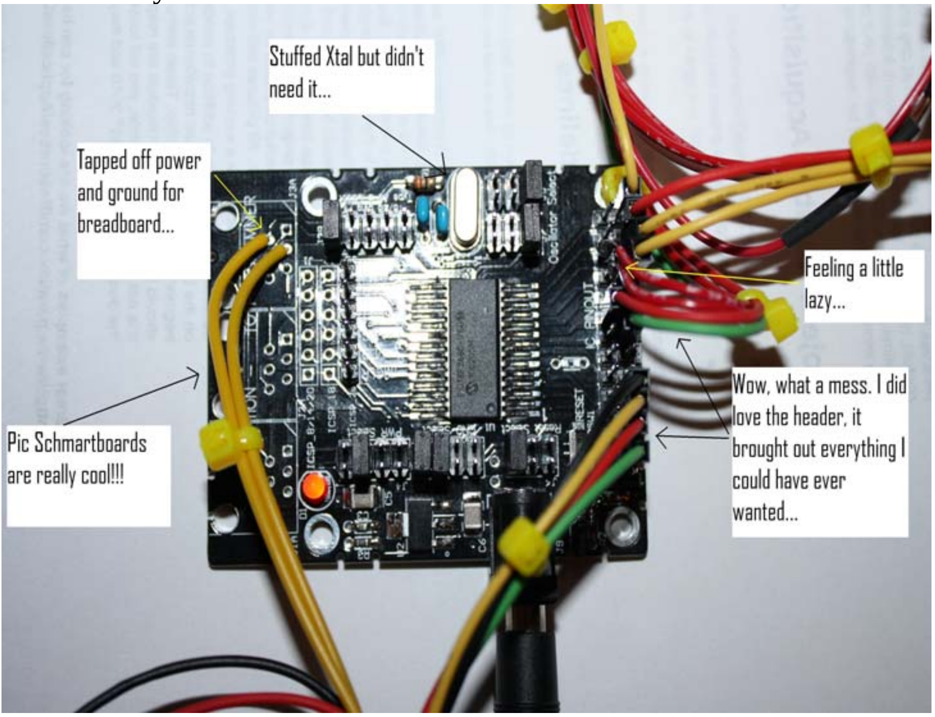
Unsolicited E-Mail textbox this box is populated with a message when my simulated alarm switch is activated

File based messaging system is used as follows:

Micro/Xport<->VB App<->VBA Macro

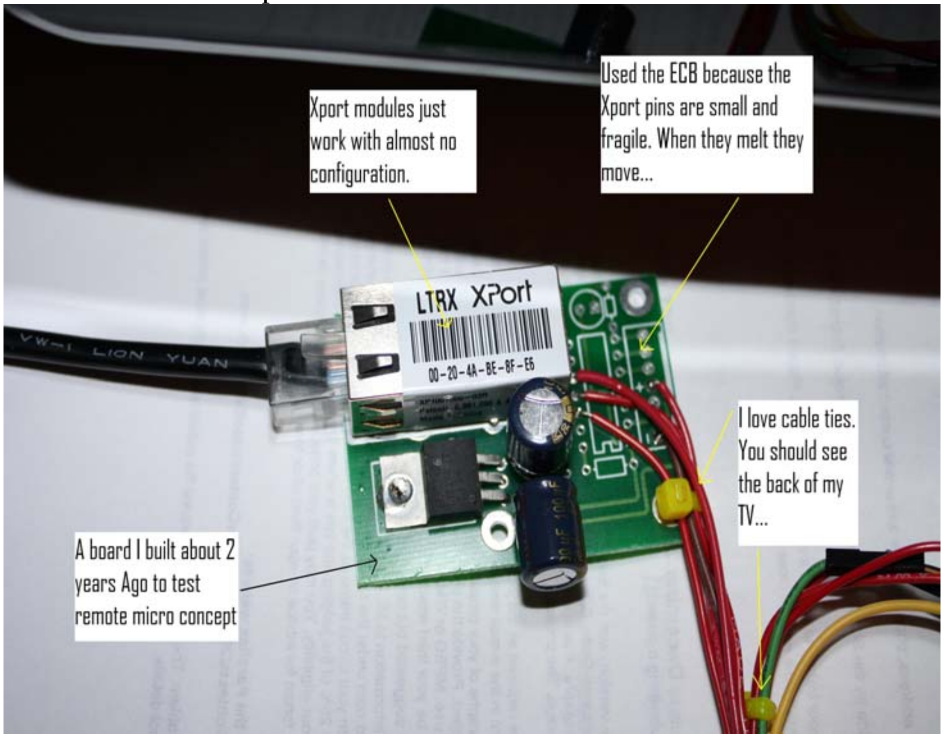
E-Mail Client Outlook 2003

Picture of my Schmartboard:


Stuffed Xtal but didn't need it...
Tapped off power and ground for breadboard...
Feeling a little lazy...
Wow, what a mess. I did love the header, it brought out everything I could have ever wanted...
Pic Schmartboards are really cool!!!

.

Closer look at the Xport ECB:


Xport modules just work with almost no configuration.
Used the ECB because the Xport pins are small and fragile. When they melt they move...
LTRX XPort
00-20-4A-BE-8F-E6
I love cable ties. You should see the back of my TV...
A board I built about 2 years Ago to test remote micro concept

Me holding my project:

Video:

I have posted a video, the resolution looked great on my PC but it looks like YouTube has done some type of compression so it is hard to make out. Luckily my commentary was great! The link is as follows:

http://www.youtube.com/watch?v=26JHZF4y5s8