Auto Dialer Wireless Smoke Detector/Alarm System

By

Manny Linhares

Senior Project

Electrical Engineering Department
California Polytechnic State University
San Luis Obispo
2010
## TABLE OF CONTENTS

List of Figures .................................................................................................................3
List of Tables ......................................................................................................................3
Acronyms ..........................................................................................................................4
Abstract .............................................................................................................................5

1. Introduction ..................................................................................................................6
2. Background ..................................................................................................................11
3. Requirements ..............................................................................................................13
4. Design ..........................................................................................................................16
5. Implementation ..........................................................................................................34
6. Testing .........................................................................................................................40
7. Hardware-Firmware Integration ...............................................................................42
8. Test Results ................................................................................................................49
9. Conclusion ..................................................................................................................53
10. Bibliography ..............................................................................................................55

Appendices ......................................................................................................................56

A. Schematics ..................................................................................................................56
B. Parts List and Cost ......................................................................................................63
C. Time Schedule Allocation .........................................................................................65
D. Basic Program listing .................................................................................................66
LIST OF FIGURES

Figure 1: Auto Dialer Wireless Smoke Detector/Alarm System ................................................. 9
Figure 2: Single RF link ........................................................................................................... 10
Figure 3: Wireless Detector/Alarm Block Diagram ............................................................... 17
Figure 4: Wireless Smoke Detector/Alarm Access Point (AP) .............................................. 21
Figure 5: Smoke Detector/Alarm Unit Firmware Flowchart .................................................. 24
Figure 6: Ionization Sensor Firmware Driver Flowchart ....................................................... 26
Figure 7: Photoelectric Sensor Firmware Driver Flowchart .................................................. 28
Figure 8: Battery Voltage Monitor Firmware Flowchart ...................................................... 30
Figure 9: AD Wireless Access Point Firmware Flowchart .................................................... 32
Figure 10: Wireless Detector/Alarm Unit. Battery, Sensor Proto board, MCU/RF Mini-board. ................................................................................................................................................ 35
Figure 11: Wireless Access Point Prototype board connected to TI’s MCU/RF Mini board and SKYLINK Auto Dialer – AD433S. ............................................................................................................. 38
Figure 12: AD Wireless Smoke Detector/Alarm Demo System .............................................. 39
Figure 13: CC1110 Mini Development Kit 868-915 MHz Content ........................................ 42
Figure 14: Ionization Sensor Interface Analog Output with no Smoke Present ...................... 44
Figure 15: Ionization Sensor Interface Analog Output with Smoke Present .......................... 45
Figure 16: Photoelectric Sensor Sampling with no Smoke Present ....................................... 47
Figure 17: Detector/Alarm Ionization Sensor Interface .......................................................... 56
Figure 18: Detector/Alarm Photoelectric Sensor Interface .................................................... 57
Figure 19: Detector/Alarm Piezo Driving Circuit ................................................................. 58
Figure 20: Detector/Alarm Power Supply and Battery Voltage Monitor ............................... 59
Figure 21: AP Piezo Buzzer Driving Circuit ......................................................................... 60
Figure 22: AP Display Driver .............................................................................................. 61
Figure 23: AP Power Supply ............................................................................................ 62

LIST OF TABLES

Table I: WIRELESS SMOKE DETECTOR/ALARM TEST RESULTS ............................................ 49
Table II: SMOKE DETECTOR/ALARM REQUIREMENTS MATRIX ........................................ 50
Table III: WIRELESS AP REQUIREMENTS MATRIX ............................................................ 51
Table IV: WIRELESS SMOKE DETECTOR/ALARM BOARD PARTS LIST .......................... 63
Table V: WIRELESS AP BOARD PARTS LIST ................................................................. 64
Acronyms

ADC  Analog to Digital Converter
AP   Access Point
DK   Development Kit
IC   Integrated Circuit
IR   Infrared
ISM  Industrial, Scientific and Medical
I/O  Input/Output
LED  Light Emitting Diode
LCD  Liquid Crystal Display
LPRF Low Power RF
MCU  Micro Controller
OP AMP Operational Amplifier
PCB  Printed Circuit Board
RF   Radio Frequency
SoC  System on Chip
SPI  Serial Peripheral Interface
TI   Texas Instruments
Abstract

Smoke Detectors/Alarms save the lives of over sixty percent of people each year in the United States. Most of the existing homes are still fitted with the normal smoke detectors either running out of the 120V ac wiring or out of a 9V dc battery. These alarms are perfectly fine to alert residents if there is a fire while they are home. But for people who are away from home when a fire occurs, it would be nice if they could be alerted automatically of such an occurrence and therefore be able to act upon the event by either calling the fire department, police or a neighbor to check on their house. With this in mind, the idea to build a smoke detector/alarm system that could alert a homeowner while he is away from home came about. This system is a wireless smoke detector/alarm system that will make an automatic phone call via POTS if an alarm is triggered by smoke from a fire. Upon smoke detection, the alarm will communicate the event to an Access Point wirelessly which in turn will trigger an auto dialer with preprogrammed phone numbers and a message to automatically call each of the numbers and relay the preprogrammed phone message.
1. **Introduction**

The **Auto Dialer Wireless Smoke Detector/Alarm System** consists of a base station, up to eight wireless smoke detector/alarm units, and an auto dialer that will call up to nine programmed phone numbers when there is smoke/fire in a home. The wireless link will operate in the 915 MHz, license free, ISM band and is set up in a star network topology. Figure 2-1 below illustrates the system architecture.

Great consideration will be given to cost and power consumption. Initial estimates for system component cost are as follows:

- Wireless base station – less than $10.00.
- Wireless smoke detector/alarm unit – less than $15.00 each
- Auto dialer unit – less than $20.00

The system has to have very low power consumption, especially the wireless smoke detector/alarm units since each one runs off of three 1.5 VDC alkaline batteries. Currently, the battery in non-wireless battery operated smoke detectors will last for approximately one year. The goal is to keep battery life in each of these units as close as possible to one year even with the addition of an RF transceiver.

Each wireless smoke detector/alarm unit will consist of dual smoke sensors, an alarm buzzer, a microcontroller, a very low-power RF transceiver, and three 1.5 V dc AA batteries. This unit will sample the air for smoke, and check battery
health once per minute and will relay that information to the base station. The microcontroller (MCU) is an 8051 core integrated with the RF transceiver thus lowering parts cost and component count. In addition the MCU/RF transceiver from Texas Instruments – CC1110 SOC – is a very low power RF IC specifically designed for the sub 1 GHz applications. An external whip antenna will have to replace the PCB antenna in the evaluation modules used for the prototype. This will extend the range of the units so that they can be placed anywhere in a normal single family home.

The base station will support up to eight wireless smoke detector/alarm units, assigned via a keypad and will scan each of the smoke detector units once per minute for information such as battery life and alarm status. It will also have remote control of each detector/alarm unit hush and battery test functions. The information received will be stored and upon manual command displayed in a small LCD display. The MCU/RF transceiver is the same used in the smoke detector/alarm units. It will also have a local alarm buzzer which can be enabled or disabled via the keypad. Upon receiving a smoke alarm from any of the smoke detector/alarm units it will trigger the auto-dialer unit to start the automatic dial up calling sequence. It will also remotely send a “smoke/fire” command to all other smoke detector/alarm units located throughout the house which will set off their alarms. Like the smoke detector alarm units a whip antenna will also be added to the base station. The base station will run off of a 12 VDC adapter
provided by the auto-dialer. During power outages it will run off a 9 VDC backup battery also provided by the auto-dialer.

The auto-dialer unit – SKYLINK AD-433S - will call at least four pre-programmed phone numbers and deliver a pre-recorded emergency message to the called parties when triggered by the base station. It gets its power from a 12 V DC adapter and has a 9 V DC backup battery used during power outages. A future improvement to the current proposed project would be to integrate the base station unit with the auto-dialer unit further reducing cost by eliminating a keypad, an LCD display, and a power supply. Figure 1 illustrates the Auto Dialer Wireless Smoke Detector/Alarm System.
Figure 1: Auto Dialer Wireless Smoke Detector/Alarm System.
A more detailed block diagram showing a single RF link is shown in Figure 2 below.

**Auto Dialer Wireless Smoke Detector/Alarm**

*Single RF Link Block Diagram*

---

*Figure 2: Single RF link.*
2. Background

The use of early warning fire and smoke detection system results in significant reduction in fire deaths each year. The sooner a fire is detected, the better the outcome for saving lives. However there are occasions when no one is at home when fire occurs and the alarm sounds. In these less frequent cases – possibly caused by short-circuits due to faulty wiring, lightning, etc. - it is highly probable that when the fire personnel arrive at the scene the home may have been totally destroyed, especially if the neighbors do not hear the fire alarm and therefore do not call the fire department promptly.

To help solve this problem, I propose to build a wireless smoke and fire detector system with an automatic dialing function. At the first sign of smoke or fire, the system will automatically dial up to four pre-programmed phone numbers and will play a pre-recorded fire emergency message alerting the called parties of the emergency. Although fire departments do not respond to “robot” calls, anyone of the called parties can call the fire department – after verifying the veracity of the emergency – and thus have a better chance of saving at least part of his or her home.

There are currently professional home security systems that perform this function but are expensive and require a monthly service fee. There are also wireless smoke detectors/alarms in the market that perform an interconnect function – each smoke detector/alarm communicates wirelessly with the all other smoke detectors/alarms in the home in a P2P (Peer-to-Peer) fashion when there is smoke or fire. But these systems are still geared to alert residents present at the time of the fire. The system I
am proposing above is to have a small, very low cost base station - the system central control unit - that communicates with all other smoke detectors/alarms in the home in a star network topology similar to the popular Wi-Fi 802.11a/g network. The base station will support up to eight wireless smoke detectors/alarms positioned in different locations of the house. The system I am proposing is intended only for single family homes and can operate either as standalone or as part of home automation central control panel.
3. **Requirements**

**Specifications**
The following paragraphs list the specifications for the three main components of the system. Note that the **Auto-Dialer** unit is an existing standalone unit which will interface to the AP/Base Station.

**Smoke Detector/Alarm Unit**

- Dual Photoelectric/Ionization smoke/fire detector sensors
- 85 decibel alarm buzzer
- Hush function/button to silence nuisance alarms
- Test function/button and LED to verify battery and alarm operation
- Warning Chirp for low battery replacement
- RF Transceiver: 915 MHz ISM Band
  
  Modulation type: FSK/FH
- Data Rate: <= 1200 bps
- RF Output Power: up to +10 dBm
- Range: Has to at least cover a four bedroom/3000 sq. ft. home, very dependent on SNR at receiver input
- Antenna Type: External Whip Antenna (omnidirectional)
- Antenna Gain: TBD (up to 6 dBi)
- Antenna Size: TBD (possibly up to 3 ½ inches)
- Power Consumption: TBD (very low)
• RF Duty Cycle (Radio On Time/(Radio On + Radio Off Time) Very Low for power savings
• Receiver Sensitivity -110 dBm @ 1200 bps
• Channel number selection 3 position DIP switch
• Power Supply 9 V DC Alkaline Battery (battery must last for approximately one year)

AP/Base Station Unit
• Monitor Detector/Alarm Units for Smoke/Fire Detection
• Signal Auto Dialer Unit upon receipt of an alarm
• Monitor Detector/Alarm Units Low Battery
• Provide Remote Controlled Detector/Alarm unit Hush function
• Provide Remote Controlled Detector/Alarm unit Battery Test function
• Sound Local Alarm Buzzer with programmable on/off capability
• LCD 1) Displays which Detector/Alarm unit(s) caused alarm
  2) Displays which Detector/Alarm unit(s) has a Low Battery
• Keypad or DIP switch to enter channel # of each active Detector/Alarm unit
• RF Transceiver Multi-Channel @ Freq = 915 MHz ISM-Band
• Modulation FSK/FH
• Data Rate <= 1200 bps
• RF Output Power up to +10 dBm (excluding Antenna)
• Antenna type External Whip Antenna (omnidirectional)
  • Antenna Gain TBD (up to 6 dBi)
- Antenna Size TBD (possibly up to 3 ½ inches)
- Power Consumption TBD (low – unit will run off DC adapter)
- Range Has to at least cover a four bedroom/3000 sq. ft. home, very dependent on SNR at receiver input
- Receiver Sensitivity -110 dBm @ 1200 bps
- Power Supply 12 VDC Adapter with 9 VDC Alkaline Battery backup. (Tapped off of Auto Dialer power supply)

**Auto Dialer Unit – Skylink Dial-Alert (Model AD-433S)**

- Program up to 9 telephone numbers to automatically dial
- Record Emergency Dial message up to 40 seconds long
- Playback recorded message
- Programmable Time Clock
- LCD
- Keypad (16 Keys) with Backlight
- 3 LED’s : ACPWR, LOBATT, DISARM
- Speaker
- Power Supply 12 VDC AC adapter, 9 VDC Battery backup
4. Design

4.1. System Design

The proposed system as shown in figure consists of an Access Point (AP) and up to eight wireless alarms. For the purposes of this project only two will be shown. This system will operate in the 915 MHz ISM band and is designed for very low power consumption in the battery operated detector/alarm units. The AP is always on and is powered off of 12Vdc from the ADS-433 auto dialer which contains a 9 VDC battery backup.

4.2. Hardware Design

4.2.1. Detector/Alarm Hardware Design

Each detector/alarm unit consists of two smoke sensors – ionization and photoelectric – a piezo transducer (buzzer), a battery voltage monitor and a battery operated power supply. Its 4.5 VDC power is supplied by three AA alkaline batteries and its wireless and microcontroller functions are provided by TI’s CC1110 SOC part of the CC1110 Mini Development Kit. A block diagram of the detector/alarm unit is shown in Figure 3.
4.2.2. Ionization Sensor Interface

The ionization sensor consists of an ionization chamber with a voltage applied between its plates and a very small amount of americium-241; a radioactive material with a half-life of 432 years that releases alpha particles that will ionize the oxygen and nitrogen atoms of the air in the chamber. The free electrons will be attracted to the plate with the positive voltage or higher potential whereas the positive ions will flow to the plate with the negative voltage or at lower potential. This creates a very small current – in the Pico-amperes range - that is detected by the electronics designed to read the sensor. The
presence of smoke in the chamber disturbs this process by neutralizing the ionization of the air atoms and thus reducing the current. This current reduction is sensed by the current-to-voltage converter designed to sense this change and will therefore trigger a smoke alarm.

A current-to-voltage converter (trans-impedance) amplifier with a high gain followed by a difference amplifier is used in this design to interface to the ionization sensor. Some reverse engineering was done because there is no datasheet freely available on such sensor due to the radioactive material included. The data for this sensor must be obtained from the NRC and smoke alarm manufacturers that use this type of sensor must obtain a license to use this material.

### 4.2.3. Photoelectric Sensor Interface

This interface supplies current to an infrared LED and reads the output of an infrared detector using a current to voltage converter with high gain. Both components are inside a dark smoke chamber to prevent interference from daylight. The theory of operation of this sensor is as follows: The IR emitter sends a beam across the chamber that will be missed by the detector in normal conditions since it is at a 40 degree angle from the line of the beam. Smoke particles, when present, will deflect part of the IR beam into the detector which will increase its current output and therefore cause an increase in voltage at
the output of the Trans impedance amplifier. This sensor is better at sensing smoldering fires than the ionization sensor.

4.2.4. Piezo (Buzzer) Interface

The Piezoelectric buzzer interface will drive a high output alarm transducer that will alert residents upon the detection of smoke. This transducer operates at 9 VDC and oscillates at approximately 3.4 KHz. This requires a step-up DC to DC converter since the detector/alarm maximum supply voltage is 4.5 VDC.

4.2.5. Battery Voltage Monitor

The battery voltage monitor circuit divides the 4.5 VDC by three and supplies it to an A/D for conversion and monitoring. It is important to note here that this monitoring circuit takes as its input the 4.5 VDC coming from the batteries and not the 3VDC regulated output.

4.2.6. Power Supply

The power supply takes 4.5 VDC from three AA alkaline batteries and regulates the voltage down to 3 VDC. This voltage supplies the Mini-Development board as well as the photoelectric sensor and is used as a reference to the photoelectric as well as the ionization sensor analog circuits. It can supply up to 250 milliamps even though the current consumption should be a lot lower.
4.2.7. **AP Hardware Design**

The main function of the AP is to receive messages from each of the smoke detector units and take the appropriate action after decoding the detector status byte that arrives in the packet payload. When any of the status alarm bits received from a detector/alarm unit is set, the AP will activate a relay that will trigger the Auto Dialer to make a phone call to all the preprogrammed phone numbers in its memory. The AP will also turn on its buzzer until such time as the alarm status bits received are all clear. As a bonus a write-only LCD interface is provided that will display the battery voltage of each of the associated smoke detectors. Figure 4 below is a block diagram of the wireless smoke detector AP.
4.2.8. Auto-Dialer Interface

The Auto-Dialer interface is a very simple interface that triggers the Auto Dialer machine by removing a short between two of its pins. This is accomplished with a normally closed relay.

4.2.9. Piezo (Buzzer) Interface

The AP piezo buzzer interface is different than the one in the smoke detectors since the AP is always on and is powered by 12 VDC. There is no need to step-up the voltage to drive the transducer. This interface regulates the 12 VDC down to 9VDC to drive the buzzer directly.
4.2.10. LCD Screen Interface

The CC1110 SPI interface is setup to drive an 8 char by 2 line small LCD. This LCD is used mainly to display the system logo as well as display each detector battery voltage.

4.3. Firmware Design

The firmware design consists of two main programs; one to drive the AD Detector/Alarm and the other to drive the AP. These programs are based on the “AP_as_data_Hub” example that comes with the TI CC1110 Mini Development Kit. They provide the wireless protocol with API calls that the customer can use to develop his application. The following sub-sections will provide the firmware design in more detail.

4.3.1. Detector/Alarm Firmware Design

The AD Wireless Smoke Detector/Alarm driver performs the following functions:

- Initializes the SimpliciTI protocol stack and joins the AP
- Establishes a link to the AP for normal communications
- It goes to sleep for 10 seconds to save power
- Upon wake up, it reads the output of both ionization and photoelectric sensors and compares the digitized results with the respective thresholds in order to make a decision about the presence of smoke
• If smoke is detected, then an averaging process occurs where three consecutive smoke detections cause an alarm which will enable a local buzzer as well as send a message to the AP with the alarm flags set.

• It reads the battery voltage every minute and generates a chirp if voltage falls below a set threshold. This number is also sent to the AP for display.

Figure 5 is a detailed flowchart of the firmware required to make the smoke detector/alarm fully functional.
Figure 5: Smoke Detector/Alarm Unit Firmware Flowchart
4.3.2. **Ionization Sensor Driver**

The ionization sensor driver enables the sensor bias voltage, sets up the A/D converter in the MCU and reads the output of a difference amplifier. It compares the result against a preset threshold and asserts an ionization alarm flag if smoke is detected. See Figure 6 for a detailed description of the algorithm.
4.3.3. Photoelectric Sensor Driver

The photoelectric sensor driver is more involved than the others. First it performs a ‘dark’ sampling – sampling of the dark chamber without enabling the IR emitter. Then it performs a ‘light’ sampling – sampling of the smoke chamber after enabling the IR emitter and transmitting an IR beam across it. Then it computes the difference of
both sampling results and compares it against a preset threshold setting a photo sensor alarm flag if smoke is present. A detailed flowchart of this routine follows in Figure 7.
Figure 7: Photoelectric Sensor Firmware Driver Flowchart
4.3.4. **Piezo (Buzzer) Driver**

This driver sets up an MCU timer and its interrupt to generate a 1 second 50% duty cycle (500 milliseconds on/ 500 milliseconds off) alarm sound upon the activation of any of the alarm flags.

4.3.5. **Battery Voltage Monitor Driver**

The Voltage Monitor Driver enables the hardware to read the battery voltage and if below calculated threshold generate an 8 millisecond chirp and assert the battery voltage low flag. Figure 8 shows a detailed design of the battery voltage firmware driver.
4.3.6. AP Firmware Design

The Access point firmware initializes the application I/O, LCD, Network, and MRFI interfaces and then listens for requests from detectors/alarms wanting to join the network. It also checks for messages from the detectors already associated with the network and processes those messages. It decodes the alarm status payload byte for any alarm bits that may be set and if so enables the relay that triggers the Auto Dialer
to make a phone call. It also enables its buzzer and displays each detector voltage on the LCD screen.
Figure 9: AD Wireless Access Point Firmware Flowchart

4.3.7. Auto-Dialer Interface Driver

This driver basically enables/disables the normally closed relay that triggers the Auto Dialer.

4.3.8. Piezo (Buzzer) Interface Driver

This driver is a copy of the detector/alarm driver described above in 4.3.4 above.
4.3.9. LCD Interface Driver

These routines were imported from

“http://www.circuitsathome.com/mcu/interfacing-lcd-via-spi” and

modified to work with the 8051 MCU since they were designed for

PIC microcontrollers.
5. Implementation

The system implementation was challenging to say the least. Two Detector/Alarm prototype Vector boards and one AP prototype board were built. It took approximately eight to ten hours to build each of the detector/alarm prototype boards and about half as much to build the much simpler AP board. Figure 10 shows the completed detector/alarm Vector board hooked up to a battery on the lefty and the Texas Instruments CC1110 Mini RF board on the right.
Figure 10: Wireless Detector/Alarm Unit. Battery, Sensor Proto board, MCU/RF Mini-board.

Refer to Appendix B “AD Smoke Detector/Alarm schematic” for the following explanations.

Ionization sensor interface implementation – A dual MCP6042 rail-to-rail OP Amp was chosen for its very low input offset current of 1 Pico Ampere and its low quiescent current of 600 Nano Ampere. This is important due to low power nature of this application and the very low current generated by the Ionization sensor. The first stage of the sensor conditioning circuit is a current-to-voltage converter with an output equal to: $V_o = (R)I_{in}$ since the sensor output pin is connected to the positive input of the operational amplifier. This stage is followed by a difference amplifier (other ½ of MCP6042) that provides the output for A/D conversion. This amplifier is biased at 1.5 VDC; 3V regulated/2. It has an output voltage given by

$$V_o = \left(1 + \frac{R2}{R1}\right)\left(\frac{R4}{R3 + R4}\right)V_2 - \left(\frac{R2}{R1}\right)V_1$$
where R3 and R4 form the reference divider into the positive input of the Op Amp and R1 and R2 are hooked up to the negative input. V1 is the output of the preceding stage I-to-V converter. As V1 decreases due to smoke detection in the ionization chamber Vo increases since the first term of the above equation is a constant; V2 = reference voltage = regulated 3 VDC. Vo is digitized by the CC1110 A/D converter and the result compared against a threshold of approximately 2 VDC. Before this circuit is sampled by the A/D an increase in bias voltage of the chamber is enabled which causes VO to be around 0.95 VDC with no smoke present. If there is smoke present then VO will increase pass its typical value of 1.28 Volts?

Photoelectric sensor interface implementation – A MCP6021 OP Amp was chosen for its very low typical input offset current of 1 Pico Ampere and a reasonably high slew rate of 7V/us with a settling time of typically 250 ns. This is important because in this implementation the Op Amp supply is provided by an MCU I/O pin which turns it off to save power since its quiescent supply current is 1 milliamp typical. When turning it on for sampling the chamber for smoke, a fast settling time keeps the firmware from waiting too long and thus save power. A quiescent current difference of 953 micro amperes was measured when this Op amp was supplied with regulated 3 VDC. Therefore the decision to use a microcontroller I/O pin to shut it off when not sampled. A transistor switch keeps the IR emitter off until sampling time. During “light” sampling this transistor switches on and a current of approximately 100 milliamps flows through the IR emitter to generate a beam in the chamber. The Op amp is biased at 0.766 volts and is gain compensated with a 5 pF capacitor. This is the typical output voltage with no smoke in the chamber. In the presence of smoke this voltage will increase to a value dependent on the reflections that occur inside the chamber and how much the IR detector will see. A photocurrent in the range of 40 to 60 micro amperes will be generated and converted to a voltage by the current-to-voltage converter.

Piezoelectric transducer interface implementation – A Microchip DC to DC up converter and Piezoelectric Horn Driver - RE 46C117 - was chosen for its low quiescent current of 7 micro amps and for its low voltage supply range of 2 to 5 Volts appropriate for battery powered systems. The step up voltage converter will output a typical 9.7 Volts to drive the buzzer.

Battery voltage monitor implementation – This circuit divides the 4.5 VDC battery voltage by 3 using a transistor (2N3904) switch to keep the circuit from wasting current when not sampled. Once a minute the MCU will turn the transistor on and sample its emitter voltage. After it is digitized a comparison is made against a
threshold of 1.10 volts arrived at by taking into account the minimum required input voltage to the low dropout regulator MCP1700 - $Vin \geq (VR + 3.0\%) + V_{dropout}$ where $VR = 3.0$ VDC and $V_{dropout} = 178$ mv. $Vin \geq 3.628$ Volts. $Vin/3 = 3.628/3 = 1.089$ V rounded up to 1.10 Volts.

Power Supply implementation – A Microchip MCP1700 LDO voltage regulator was chosen for its 1.6 micro amps typical quiescent current. The regulator supplies 3 VDC to most of the analog circuits on board and the CC1110 Mini RF board.

Figure 11 shows a fully implemented AP Vector board connected to a TI CC1110 Mini MCU/RF Development board and Auto Dialer.

Display interface circuit implementation – this circuit consists of a 74HC495 8-bit serial to parallel shift register to drive an 8x2 LCD display. Both operate at regulated 3 VDC.

Auto Dialer interface implementation – The MCU turns on a transistor switch which turns on a solid state relay – TLP4227G to trigger the automatic phone call in the Auto Dialer.

Power Supply implementation – The AP gets its 12 VDC supply from the Auto Dialer and is regulated down to 3 VDC using an LM317LZ. The AP is always on to listen to possible messages from the detectors/alarms thus power consumption will be higher.

Piezoelectric buzzer implementation – same as in the detectors.
Figure 11: Wireless Access Point Prototype board connected to TI's MCU/RF Mini board and SKYLINK Auto Dialer – AD433S.

Figure 12 shows the full demo system implemented.
Figure 12: AD Wireless Smoke Detector/Alarm Demo System.
6. Testing

The testing of the hardware was done using a divide and conquer approach due to the many components of the system. Each circuit was tested and debugged in a breadboard before its implementation in the final prototype Vector board. The firmware was tested during the hardware-firmware integration phase. The following test plan was followed to verify the design of each circuit:

- Test Ionization Sensor interface difference amplifier for an output of 2.325 VDC with V1 input at 0 volts.
- Test Ionization Sensor interface difference amplifier for an output of 0.825 VDC when V1 input is 3VDC.
- Test Ionization sensor interface first stage current-to-voltage converter.
- Perform a “light” test and a “dark” test on the photoelectric sensor interface.
- Repeat same tests with smoke present.
- Test Piezo transducer driver circuit for a loud sound at 3.4 KHz +- 500 Hz.
- Test Battery Voltage Monitor circuit for 4.5 VDC/3 output when enabled and approximately 0 VDC when disabled.
- Find smoke detected thresholds for both Ionization and Photoelectric interface circuits by performing such tests with and without smoke present.
• Test phone call triggering relay in AP for an open when enabled. Otherwise it shall be a closed circuit with approximately 15 Ohm impedance.

• Test AP LCD display interface by displaying “AD SmkDet-Alrm System” in 2 lines of 8x2 LCD display.

• Perform a System Test – introduce smoke to Detector/Alarm and verify that local buzzer goes off, message is sent to AP and a phone call is triggered on the AD-433 auto dialer and the local AP buzzer is turned on until alarms are clear.

• Test RF link using TI’s packet sniffer and verify that Detector/Alarm status is getting to Access Point.

• Test RF Transceiver Range.
7. **Hardware-Firmware Integration**

The hardware–Firmware integration was the hardest part of the project. It is at this phase that all the parts come together. The following tools were used for the hardware-firmware development:

- Texas Instruments. “CC Debugger”
- Texas Instruments. “SmartRF Studio7”
- Texas Instruments. “SmartRF Flash Programmer”
- Texas Instruments. “Packet Sniffer”

Figure 13 shows the TI Mini Development Kit used in this project.

![Figure 13: CC1110 Mini Development Kit 868-915 MHz Content](image-url)
Each of the TI SmartRF CC1110 target boards came programmed to operate in the European 868 MHz band. Some of the RF parameters such as frequency and data rate were changed to operate at the US 915 MHz ISM band and 2.4 KBAUD respectively.

During this phase all the analog measurements taken were digitized by an on chip ADC with a reference voltage of VDD= 3VREG on its AVDD pin and a resolution of 12 bits in a 2’s complement format. The conversion rate

\[ T_{\text{conv}} = (\text{decimation rate} + 16) \times T, \text{where } T \leq 0.23 \text{ usec} \]

The ADC runs at 4.33 MHz: system clock = 26MHz/6. A 512 decimation rate was chosen for higher accuracy and the conversion rate was 121.44 microseconds. Refer to the CC11100F8 datasheet section 12.10 for more detailed information on the ADC.

Figure 14 below shows the analog output of the Ionization Sensor interface circuit with no smoke present. Channel 2 (green trace) shows the output voltage at approximately 1.18 VDC. This implies an ionization sensor current of approximately 2.08 microamperes with no smoke present. Channel 1 (yellow trace) shows the circuit is enabled for 1.44 milliseconds to take the measurement. This time can definitely be optimized to a much shorter period by selecting a 256 (10 bits resolution) or even 128 (9 bits resolution) decimation ratio for the ADC and by decreasing delays in the firmware “ReadIonSensor()” routine.
Figure 15 below shows the analog output of the ionization sensor interface circuit with some smoke present. Channel 2 (green trace) shows an output voltage increase of approximately 220 millivolts which implies an ionization sensor current of approximately 1.68 microamperes. This is in line with theory which states that when smoke particles are present in the ionization chamber the process of ionization is disrupted and the current inside the chamber decreases. This decrease is difficult to quantify theoretically without a precise knowledge of the quantity of smoke that entered the chamber and the lack of a datasheet on this sensor for reasons mentioned.
previously. A threshold of 1350: $1350 \times \frac{3}{2047} = 1.98 \text{ Volts}$ was chosen but during the presentation it had to be lowered to 1000: $1000 \times \frac{3}{2047} = 1.46 \text{ Volts}$ due to the difficulty in getting smoke into the chamber in open air without a detector/alarm enclosure.

Figure 15: Ionization Sensor Interface Analog Output with Smoke Present.

Figure 16 below shows the analog output voltage (green trace) and the sequence necessary to sample the photoelectric sensor. First the OP AMP is turned on by an MCU pin – this is done to save power. Then a “dark” sampling is taken and finally a “light” sampling is taken. The “dark” sampling is simply reading the output of the
current-to-voltage converter with the IR emitter off. This takes approximately 2 milliseconds as shown in the picture. For the “light” sampling the IR emitter is turned on for approximately 1.4 milliseconds, (shown by Channel 1 – yellow trace - enabled) and a second sample is taken. The difference of the two samples is taken and compared to a threshold to determine if there is smoke present. The voltages obtained for the “dark” and “light” sampling tests without smoke present were 0.731 VDC and 0.734 VDC respectively. With smoke present the “dark” sample voltage didn’t change whereas the “light” sample voltage went as high as $859 \times \frac{3}{2047} = 1.26$ Volts.

The decision was made on a threshold of $(light \ sample - dark \ sample) > 200, \ \Delta = \frac{200 + 3V \ ref}{2047} = 293 \ millivolts$ for the presence of smoke. During the presentation this threshold was lowered to Delta=100 due to the difficulty in getting smoke into the smoke chamber.
Figure 16: Photoelectric Sensor Sampling with no Smoke Present.

The Battery Voltage Monitor interface is enabled by firmware for 200 microseconds by setting MCU P2_0 output pin to a high and reading the emitter voltage of Q4 on P0_2. This voltage shall be equal to \( \frac{\text{VBAT}}{3} = \frac{4.5 \text{ Volts}}{3} = 1.5 \text{ VDC} \) for a pack of three new AA batteries. Based on the dropout voltage of the MCP1700 LDO regulator a low battery voltage was calculated to be 3.268 VDC. For this case a reading of \( \frac{3.286}{3} = 1.089 \text{ V} \) is read by the ADC. A threshold of 750, approximately 1.10 Volts, was programmed in firmware. A **chirp** of 8 milliseconds is generated if the read voltage is smaller than the aforementioned threshold.
The buzzer firmware driver generates a 500 MS On, 500 MS off sound by enabling/disabling MCU pin P0_7. The buzzer hardware circuit oscillates at approximately 3.4 KHz.

The hardware-firmware integration of the AP consisted of driving the buzzer as in the Detector/Alarm board, processing the messages received from both Detector/Alarm boards, and opening the normally closed relay IC - TL4227G, using MCU pin P1_4, when an alarm condition is received by the AP. In addition, the battery health status of each of the two detector/alarms was displayed on the small LCD as a voltage.
8. Test Results

Table I shows test results of Detector/Alarm board #1 subsystem during hardware-firmware integration.

Table I: WIRELESS SMOKE DETECTOR/ALARM TEST RESULTS

<table>
<thead>
<tr>
<th>Interface</th>
<th>Conditions</th>
<th>ADC Values</th>
<th>Output Voltage (Volts)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>W/O Smoke</td>
<td>W/Smoke</td>
</tr>
<tr>
<td>Ionization Sensor</td>
<td>Chamber Bias 2.56V</td>
<td>673-944</td>
<td>1296-1396</td>
</tr>
<tr>
<td>Photoelectric Sensor</td>
<td>Dark Test IR OFF</td>
<td>499-501</td>
<td>499-501</td>
</tr>
<tr>
<td>Photoelectric Sensor</td>
<td>Light Test IR ON</td>
<td>501-502</td>
<td>600-859</td>
</tr>
<tr>
<td>Battery Voltage Monitor</td>
<td>Q4 ON Batteries at 4.36 V</td>
<td>1001</td>
<td>NA</td>
</tr>
</tbody>
</table>

The Piezo sounder was tested and produced a “loud” sound with a measured frequency of approximately 3.6 KHz. This conforms to the datasheet specification of 3.4 KHz ± 500 Hz.

The voltage regulator output was 3.007 VDC from a battery input of 4.5 VDC. As the project progressed and the battery voltage decreased to 3.28 VDC, the output of the regulator maintained its voltage at 3.0 VDC as expected.

The measured values of the Access Point circuits were as follows:

Supply quiescent current: 5.5 milliamps.

Voltage regulator output: 3 VDC.

Solid State relay normally closed (Form 1B) ON resistance: 14.8 ohms versus specification of 15.0 ohms.
Verified that with P1_4 at 3 VDC – relay enabled – the contacts were open, condition that triggers the Auto Dialer to make a phone call.

Performed an **RF Link Test** with the CC1110 Mini-Development target boards from TI by leaving one of the transceivers in the Senior Projects Lab. And walking the other one until it failed (red LED on). I was able to walk to Lucy’s café next to the Student Union before the first packet loss occurred. This represents a distance of approximately 250 yards. This range is more than enough to meet one of the project requirements of placing the detectors/alarms in a four bedroom/3000 sq. feet house. Due to this result, I decided that a whip antenna was no longer needed to demonstrate this project and that the current PCB etched antenna was perfectly fine. In addition this antenna is much cheaper than the whip antenna.

Table II below is a matrix that shows the project requirements that were met and the ones that were not met for the Wireless Smoke Detector/Alarm.

**Table II: SMOKE DETECTOR/ALARM REQUIREMENTS MATRIX**

<table>
<thead>
<tr>
<th>REQUIREMENT</th>
<th>MET</th>
<th>NOT MET</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual Photoelectric/Ionization smoke/fire detector sensors</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85 decibel alarm buzzer</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Function/Button and LED to verify Battery and Alarm Operation</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Warning Chirp for Low Battery replacement | Y | | 
| Low Power Consumption | Y | This assertion solely based on quiescent current measured during board bring up and Battery voltage drop from 4.5V to 3.8V for the 3 weeks project duration. A current profile is still needed to better assess this parameter. |
| Hush function/button to silence nuisance alarms | X | Due to lack of time |
| Send messages to AP every 10 seconds with Alarm and Battery info. | Y | |

Table III below is a matrix that shows the project requirements that were met and the ones that were not met for the Wireless AP.

**Table III: WIRELESS AP REQUIREMENTS MATRIX**

<table>
<thead>
<tr>
<th>REQUIREMENT</th>
<th>MET</th>
<th>NOT MET</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Auto Dialer Unit upon receipt of an alarm from detector/alarm units.</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor Detector/Alarm units for low battery.</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote controlled Hush function.</td>
<td>X</td>
<td>Not implemented due to time constraints.</td>
<td></td>
</tr>
</tbody>
</table>
A test dry run of the system was done successfully the day before the presentation. An actual phone call could not be made because the AD 433 is only compatible with POTS and PABX phone lines and Cal Poly uses an All-Digital phone network. However it was observed that in fact the AD 433 attempted to make the phone call by dialing the preprogrammed phone number. It was also noticed during this test that it was difficult to get smoke into the sensor chambers. During the presentation the smoke thresholds were adjusted in firmware and the system worked as intended. A plastic enclosure would be a requirement in further development.
9. Conclusion

This was a successful project that took a little over three weeks to complete. The main idea – to make an automatic phone call when there is a fire – was clearly demonstrated with the prototype. There were many challenges such as, time constraints, learning the development tools, and hardware-firmware integration.

This project was a little too ambitious for the suggested time allotted for a senior project as can be seen by the 256 hours spent to complete it. Due to time constraints some of the original requirements were not implemented. These were the “Remote Test “, and “Remote Hush” functions.

The final cost of $26.90 for the Wireless Detector/Alarm board came fairly close to the original estimate of $20 per board. This does not include the cost of the MCU/RF target board which was $37.50 as half of the development kit (2 transceiver boards) cost. The transceiver Soc has a cost of $2.55 in 1Ku quantity and estimating the additional cost of the balun, crystals, and TX filter components to be $5.00 in 1Ku quantities the final cost of the Wireless Detector/Alarm Board should be close to $35. Of course this cost could possibly be cut in half for quantities of 100 K units or above. The cost of the Wireless AP board was $27.68 quite higher than the original estimate of $10. This is in part due to the LCD display cost of $8.25 and the proto board cost of $5.75. These costs would be assimilated into a single cost in a future development where the AP would be integrated in the Auto Dialer thus saving duplicate parts costs.
There is still room for improvement and in a future development an important interconnect function should be implemented. This function shall reside in the AP which upon receiving an alarm from any of the networked detectors shall broadcast a message to all other detector units in the network to turn their buzzers on. This would require the RF transceiver in the detector units to operate either in polled mode or for it to stay awake for a predefined amount of time to be able to receive the message broadcast by the AP repeatedly. The tradeoff would be an increase in power consumption for the detector units. The next step to possibly make a viable commercial unit out of this system would be to integrate the MCU/RF and the Detector/Alarm boards into a single PCB, and optimize the firmware for lower power consumption. The AP board, its MCU/RF board, and the AD 433 should also be integrated into one single PCB to eliminate duplicate components and therefore save on cost.

In conclusion this was a successful and fun project that showed the realization of an idea from concept to finished prototype in a record amount of time.
10. Bibliography


Texas Instruments. “CC1110 Mini Development Kit 868/915 MHz”,
http://focus.ti.com/docs/toolsw/folders/print/cc1110dk-mini-868.html

“LCD Interface Schematics and Software routines”
http://www.circuitsathome.com/mcu/interfacing-lcd-via-spi

“Ionization and Photoelectric Sensors Theory of Operation”
http://home.howstuffworks.com/home-improvement/householdsafety/fire/smoke.htm

“SimpliciTI Compliant Protocol Stack”
http://focus.ti.com/docs/toolsw/folders/print/simpliciti.html

*SimpliciTI Specification Version 1.1.1*, Texas Instruments, 2010

*SimpliciTI Application Programming Interface*, TI Document Number: SWRA221

*SimpliciTI Sample Application Guide*, TI Document Number: SWRA243

Texas Instruments, “Implementing A Smoke Detector With The MSP430F2012”,
Application Report: SLAA335-October 2006
Appendices

A. Schematics

AD Wireless Detector/Alarm

Figure 17: Detector/Alarm Ionization Sensor Interface.
Figure 18: Detector/Alarm Photoelectric Sensor Interface.
Figure 19: Detector/Alarm Piezo Driving Circuit.
Figure 20: Detector/Alarm Power Supply and Battery Voltage Monitor.
AD Wireless AP

Figure 21: AP Piezo Buzzer Driving Circuit.
Figure 22: AP Display Driver.
Figure 23: AP Power Supply.
### B. Parts List and Cost

#### AD Wireless Smoke Detector/Alarm Board

#### Table IV: WIRELESS SMOKE DETECTOR/ALARM BOARD PARTS LIST

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Description</th>
<th>Quantity</th>
<th>Value</th>
<th>Ttl Qty</th>
<th>Cost/unit</th>
<th>Ttl Cost</th>
<th>Vendor</th>
<th>Vendor Part#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.3PF 61A AVX Corporation</td>
<td>1</td>
<td>$0.24</td>
<td>2</td>
<td>$0.48</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1N4148 DIODE SWITCHING</td>
<td>2</td>
<td>$0.14</td>
<td>2</td>
<td>$0.28</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1N5817 DIODE SCHOTTKY</td>
<td>1</td>
<td>$0.54</td>
<td>1</td>
<td>$0.54</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1N4003-T RECTIFIER</td>
<td>1</td>
<td>$0.32</td>
<td>1</td>
<td>$0.32</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MCP6042-I/P IC OPAMP</td>
<td>1</td>
<td>$0.96</td>
<td>1</td>
<td>$0.96</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MCP6021-E/P IC OPAMP</td>
<td>1</td>
<td>$1.12</td>
<td>1</td>
<td>$1.12</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MCP1700-3002E/TO IC REG LDO</td>
<td>1</td>
<td>$1.38</td>
<td>1</td>
<td>$1.38</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>P2N2222AG TRANS GP SS NPN LP</td>
<td>2</td>
<td>$0.53</td>
<td>2</td>
<td>$1.06</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2N3904G TRANS SS NPN GP</td>
<td>1</td>
<td>$0.51</td>
<td>1</td>
<td>$0.51</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>PORT14 HDR2X14 Molex Connector</td>
<td>1</td>
<td>$1.50</td>
<td>1</td>
<td>$1.50</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>SBH-331AS BATTERY HOLDER</td>
<td>1</td>
<td>$1.30</td>
<td>1</td>
<td>$1.30</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CT2063-3 DIP SWITCH</td>
<td>1</td>
<td>$0.76</td>
<td>1</td>
<td>$0.76</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>V2010-ND PROTOBOARD</td>
<td>1</td>
<td>$7.33</td>
<td>1</td>
<td>$7.33</td>
<td>47.8169</td>
<td>LGI</td>
<td></td>
</tr>
</tbody>
</table>

**Average cost per board $26.90**

**Total cost $26.90**

---

**Footnotes:**

- **CAPACITORS**
  - # boards to build => 1

- **INDUCTORS**
  - # boards to build => 1

- **RESISTORS AND POTS**
  - # boards to build => 1

- **DIODES**
  - # boards to build => 1
## AD Wireless AP Board

### Table V: WIRELESS AP BOARD PARTS LIST

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Line #</th>
<th>Qty</th>
<th>Reference</th>
<th>Part Name</th>
<th>Value</th>
<th>Tol</th>
<th>Mfg</th>
<th>Mfg Part#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2.1UF</td>
<td>±10%</td>
<td>AVX Corp.</td>
<td>CAP CER</td>
<td>2</td>
<td>0.50</td>
<td>0.40</td>
<td>476-1317-AD</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>10UF</td>
<td>±10%</td>
<td>TDK Corp.</td>
<td>CAP CER</td>
<td>1</td>
<td>0.83</td>
<td>0.83</td>
<td>445-2817-AD</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1000PF</td>
<td>±5%</td>
<td>AVX Corp.</td>
<td>CAP CER</td>
<td>1</td>
<td>0.51</td>
<td>0.51</td>
<td>476-2817-AD</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1UF</td>
<td>±10%</td>
<td>AVX Corp.</td>
<td>CAP TANTALUM</td>
<td>1</td>
<td>0.46</td>
<td>0.46</td>
<td>476-1317-AD</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>10UH</td>
<td></td>
<td>MURATA</td>
<td>INDUCTOR</td>
<td>1</td>
<td>0.60</td>
<td>0.60</td>
<td>811-2040-ND</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1.0KOHM</td>
<td>±5%</td>
<td>PANASONIC-ECG</td>
<td>RES CARBON FILM</td>
<td>1/4W</td>
<td>2</td>
<td>0.09</td>
<td>0.18</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>249 OHM</td>
<td>±2%</td>
<td>PANASONIC-ECG</td>
<td>RES CARBON FILM</td>
<td>1/4W</td>
<td>1</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>348 OHM</td>
<td>±2%</td>
<td>PANASONIC-ECG</td>
<td>RES CARBON FILM</td>
<td>1/4W</td>
<td>1</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1.5MOHM</td>
<td>±1%</td>
<td>STACKPOLE</td>
<td>RES MF</td>
<td>1/4W</td>
<td>1</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>150KOHM</td>
<td>±1%</td>
<td>PANASONIC</td>
<td>RES METAL FILM</td>
<td>1/4W</td>
<td>1</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>10.0KOHM</td>
<td>±5%</td>
<td>PANASONIC-ECG</td>
<td>RES CARBON FILM</td>
<td>1/4W</td>
<td>7</td>
<td>0.09</td>
<td>0.63</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1.5A</td>
<td></td>
<td>FAIRCHILD</td>
<td>DIODE SCHOTTKY</td>
<td>20V</td>
<td>1</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td></td>
<td></td>
<td>MICROCHIP</td>
<td>IC HORN DRIVER</td>
<td>DUAL 8PDIP</td>
<td>1</td>
<td>0.44</td>
<td>0.44</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td></td>
<td></td>
<td>TOSHIBA</td>
<td>PHOTORELAY</td>
<td>MOSFET OUT 3MA 4-DIP</td>
<td>1</td>
<td>2.13</td>
<td>2.13</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
<td>ST MICROELECTRONICS</td>
<td>IC SHIFT REGISTER</td>
<td>8BIT 16DIP</td>
<td>1</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td></td>
<td></td>
<td>NXP SEMICONDUCTORS</td>
<td>IC SHIFT REGISTER</td>
<td>8BIT 16DIP</td>
<td>1</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td></td>
<td></td>
<td>NEWHAVEN DISPLAY</td>
<td>LCD MOD CHAR 2X8 Y/G REFLECTIVE</td>
<td>2</td>
<td>8.25</td>
<td>8.25</td>
<td>NHD-0208AZ-RN-YBW-3V-ND</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td></td>
<td></td>
<td>ON SEMICONDUCTOR</td>
<td>LED DRIVER</td>
<td>OPTO 2.8V 5V 16DIP</td>
<td>2</td>
<td>0.44</td>
<td>0.44</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td></td>
<td></td>
<td>ZETEX</td>
<td>IC TO-92</td>
<td>REGULATED</td>
<td>1</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td></td>
<td></td>
<td>NEXperia</td>
<td>IC TO-92</td>
<td>REGULATED</td>
<td>1</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td></td>
<td></td>
<td>NEXperia</td>
<td>IC TO-92</td>
<td>REGULATED</td>
<td>1</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td></td>
<td></td>
<td>NEXperia</td>
<td>IC TO-92</td>
<td>REGULATED</td>
<td>1</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td></td>
<td></td>
<td>NEXperia</td>
<td>IC TO-92</td>
<td>REGULATED</td>
<td>1</td>
<td>0.76</td>
<td>0.76</td>
</tr>
</tbody>
</table>

### Approx. cost per board: $27.68
## C. Time Schedule Allocation

<table>
<thead>
<tr>
<th>Dates</th>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/02/10 – 10/4/10</td>
<td>Project Definition and Proposal.</td>
<td>13.5</td>
</tr>
<tr>
<td>10/11/10</td>
<td>Subsystem Block Diagrams for Smoke detector/Alarm and AP.</td>
<td>2.0</td>
</tr>
<tr>
<td>10/12/10 – 10/26/10</td>
<td>Design of Wireless Smoke Detector/Alarm Interfaces.</td>
<td>20.0</td>
</tr>
<tr>
<td>10/27/10 – 10/31/10</td>
<td>Design of Access Point Interfaces.</td>
<td>8.0</td>
</tr>
<tr>
<td>11/01/10 – 11/03/10</td>
<td>Smoke Detector/Alarm and AP Schematic Entry.</td>
<td>6.0</td>
</tr>
<tr>
<td>11/04/10 – 11/6/10</td>
<td>Bill of Materials (BOMs) for Smoke Detector/Alarm and AP. Order parts.</td>
<td>12.0</td>
</tr>
<tr>
<td>11/07/10 – 11/11/10</td>
<td>Breadboard and Debug Smoke Detector/Alarm Interface Circuits.</td>
<td>22.0</td>
</tr>
<tr>
<td>11/11/10 – 11/14/10</td>
<td>Build one Smoke Detector/Alarm Board And one Access Point board.</td>
<td>25.0</td>
</tr>
<tr>
<td>11/15/10 – 11/16/10</td>
<td>Bring up and Test Smoke Detector/Alarm Board.</td>
<td>10.0</td>
</tr>
<tr>
<td>11/16/10 – 11/17/10</td>
<td>Bring up and Test AP Board.</td>
<td>3.5</td>
</tr>
<tr>
<td>11/18/10 – 11/20/10</td>
<td>System Firmware design and Development.</td>
<td>23.0</td>
</tr>
<tr>
<td>11/20/10 – 11/21/10</td>
<td>FW Implementation for Wireless Smoke Detector/Alarm.</td>
<td>10.0</td>
</tr>
<tr>
<td>11/22/10 – 11/24/10</td>
<td>Firmware/Hardware Integration and Debug of Smoke Detector/Alarm.</td>
<td>23.0</td>
</tr>
<tr>
<td>11/25/10 – 11/26/10</td>
<td>Firmware Implementation of AP.</td>
<td>10.0</td>
</tr>
<tr>
<td>11/26/10 – 11/27/10</td>
<td>FW/HW Integration and Debug of AP.</td>
<td>20.0</td>
</tr>
<tr>
<td>11/27/10 – 11/28/10</td>
<td>Build second Detector/Alarm board.</td>
<td>14.0</td>
</tr>
<tr>
<td>11/29/10 – 12/02/10</td>
<td>Write Senior Project Report.</td>
<td>18.0</td>
</tr>
<tr>
<td>12/01/10</td>
<td>Perform Dry Run of the System.</td>
<td>2.0</td>
</tr>
<tr>
<td>12/02/10</td>
<td>Presentation preparation and Presentation.</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>256</strong></td>
</tr>
</tbody>
</table>
D. Basic Program listing

**Important:** The following listing of the main code for both the Detector/Alarm and the AP is not enough to make the system functional. The following code is just the application code that runs on top of the network “NWK” and minimal RF Interface “MRFI” code. The full suite is part of SimpliciTI and can be downloaded for free from TI’s website:

The following code listing implements the main flowchart of Figure 5 above.

```
/****************************************************************************
/* AD Wireless Smoke Detector/Alarm Main Program */
/****************************************************************************
***************************************************************************/

Copyright 2007-2009 Texas Instruments Incorporated. All rights reserved.

IMPORTANT: Your use of this Software is limited to those specific rights granted under the terms of a software license agreement between the user who downloaded the software, his/her employer (which must be your employer) and Texas Instruments Incorporated (the "License"). You may not use this Software unless you agree to abide by the terms of the License. The License limits your use, and you acknowledge, that the Software may not be modified, copied or distributed unless embedded on a Texas Instruments microcontroller or used solely and exclusively in conjunction with a Texas Instruments radio frequency transceiver, which is integrated into your product. Other than for the foregoing purpose, you may not use, reproduce, copy, prepare derivative works of, modify, distribute, perform, display or sell this Software and/or its documentation for any purpose.

YOU FURTHER ACKNOWLEDGE AND AGREE THAT THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.
IN NO EVENT SHALL TEXAS INSTRUMENTS OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY
THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.

Should you have any questions regarding your right to use this Software, contact Texas Instruments Incorporated at www.TI.com.

#include "bsp.h"
#include "mrfi.h"
#include "nwk_types.h"
#include "nwk_api.h"
#include "bsp_leds.h"
#include "bsp_buttons.h"
#include "bsp_extended.h"
#include "ioCCxx10_bitdef.h"

#include "app_remap_led.h"

#ifndef APP_AUTO_ACK
#error ERROR: Must define the macro APP_AUTO_ACK for this application.
#endif

__interrupt void timer1_ISR(void);

static void linkTo(void);

void Init_8051(void);
uint16_t ReadBatVoltage(void);
uint8_t ReadPhotoSensor(void);
uint8_t ReadIonSensor(void);
static void BuzzerSetup(void);
void delay_msec(uint8_t msec);

static uint8_t sTid = 0;
static linkID_t sLinkID1 = 0;

static uint16_t batt_volt;
//static uint16_t sleep_time = 2000;
static uint16_t sleep_time = 10000;
//static uint16_t sleep_time = 100;
static uint8_t power_mode = 2;
//static uint8_t power_mode = 0;

static uint8_t smoke_count = 0;
static uint8_t bat_volt_lo = 0;
static uint8_t photo_sensor_alarm = 0;
static uint8_t ion_sensor_alarm = 0;

volatile uint8_t alarm_flag;
extern volatile uint8_t sixty_second_flag;

#define SLEEP_31_25_US_RESOLUTION 0
#define SLEEP_1_MS_RESOLUTION 1
#define SLEEP_32_MS_RESOLUTION 2
#define SLEEP_1_S_RESOLUTION 3

#define SPIN_ABOUT_A_SECOND NWK_DELAY(1000)
#define SPIN_ABOUT_5_SECONDS NWK_DELAY(5000)
#define SPIN_ABOUT_A_QUARTER_SECOND NWK_DELAY(250)

#define BAT_VOLT_TRSH 750
  //define BAT_VOLT_TRSH 950
  //define PHOTO_ALARM_THRES 200
#define PHOTO_ALARM_THRES 100
  //define ION_ALARM_THRES 1350
#define ION_ALARM_THRES 800
#define BIT0 0x01
#define BIT1 0x02
#define BIT3 0x08
#define BIT4 0x10

/* How many times to try a Tx and miss an acknowledge before doing a scan */
#define MISSES_IN_A_ROW 2

void main (void)
{
  /* Initialize 8051 for Smoke Detector App. */
  Init_8051();

  /* Initialize BSP Buttons and LEDs */
  BSP_Init();

  /* Initialize Piezo Buzzer */
  BuzzerSetup();

  /* If an on-the-fly device address is generated it must be done before the
  * call to SMPL_Init(). If the address is set here the ROM value will not
  * be used. If SMPL_Init() runs before this IOCTL is used the IOCTL call
  * will not take effect. One shot only. The IOCTL call below is conformal.
  */
  #ifdef I_WANT_TO_CHANGE_DEFAULT_ROM_DEVICE_ADDRESS_PSEUDO_CODE

  /* If an on-the-fly device address is generated it must be done before the
  * call to SMPL_Init(). If the address is set here the ROM value will not
  * be used. If SMPL_Init() runs before this IOCTL is used the IOCTL call
  * will not take effect. One shot only. The IOCTL call below is conformal.
  */
  #ifdef I_WANT_TO_CHANGE_DEFAULT_ROM_DEVICE_ADDRESS_PSEUDO_CODE


{ addr_t lAddr;

createRandomAddress(&lAddr);
SMPL_IOCTL(IOCTL_OBJ_ADDR, IOCTL_ACT_SET, &lAddr);
}
#endif /* I_WANT_TO_CHANGE_DEFAULT_ROM_DEVICE_ADDRESS_PSEUDO_CODE */

/* Keep trying to join (a side effect of successful initialization) until
 * successful. Toggle LEDs to indicate that joining has not occurred.
 */
while (SMPL_SUCCESS != SMPL_Init(0))
{
    BSP_TOGGLE_LED1();
    BSP_TOGGLE_LED2();
    SPIN_ABOUT_A_SECOND;
}

/* LEDs on solid to indicate successful join. */
if (!BSP_LED2_IS_ON())
{
    BSP_TOGGLE_LED2();
}
if (!BSP_LED1_IS_ON())
{
    BSP_TOGGLE_LED1();
}

/* Unconditional link to AP which is listening due to successful join. */
linkTo();

while (1);
}

static void linkTo()
{
    uint8_t      msg[4];
    uint8_t      noAck;
    uint8_t      batt_alarm, photo_alarm, ion_alarm;
    smplStatus_t rc;

    #ifdef APP_AUTO_ACK
    uint8_t misses, done;
    #endif
}
/* Keep trying to link... */
while (SMPL_SUCCESS != SMPL_Link(&sLinkID1))
{
    BSP_TOGGLE_LED1();
    BSP_TOGGLE_LED2();
    SPIN_ABOUT_A_SECOND;
}

/* Turn off LEDs. */
BSP_TOGGLE_LED1();
BSP_TOGGLE_LED2();

/* sleep until button press... */
SMPL_ioctl(IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_SLEEP, 0);
IEN2 &= ~0x01; // Disable RF interrupt

while (1)
{
    if (BSP_BUTTON1())
    {
        SPIN_ABOUT_A_QUARTER_SECOND; /* debounce */
P0 ^= 0x08; /* Turn on TEST function by biasing ION Chamber to to 1.25 volts */
    }
if (!alarm_flag) {  /* If there is an alarm do not go to sleep */
    // Disable Timer 1 interrupt before going to sleep
    T1IE = 0;
    BSP_SleepFor (power_mode, SLEEP_1_MS_RESOLUTION, sleep_time);
    // Re-enable Timer 1 interrupt after wake-up
    T1IE = 1;
    BSP_TURN_ON_LED1();
    SPIN_ABOUT_A_QUARTER_SECOND;
    }
else
    BSP_TURN_OFF_LED1();

    ion_sensor_alarm   = ReadIonSensor();
    photo_sensor_alarm = ReadPhotoSensor();

    /* Perform Smoke Alarm Checks */
    if (ion_sensor_alarm || photo_sensor_alarm) {
        if (!alarm_flag) {
            smoke_count += 1;
            if (smoke_count == 1)
                //sleep_time = 2500;
sleep_time = 5000;
else if (smoke_count == 2)
    //sleep_time = 1250;
    sleep_time = 1000;    // 1 second
else if (smoke_count == 3) {
    alarm_flag = 1;
    power_mode = 0;
    T1CTL |= 0x0E;   /* Enable Buzzer upon an alarm */
    BSP_TURN_ON_LED2();
    BSP_TURN_OFF_LED1();
}
}
else
    smoke_count = smoke_count;
}
else {
    alarm_flag = 0;
    smoke_count = 0;
    T1CTL &= 0xFC;   /* Disable Buzzer when alarm goes away */
    P0_7 = 0;
    power_mode = 2;
    //power_mode = 1;
    //sleep_time = 5000;
    sleep_time = 10000;
    //sleep_time = 100;
    //BSP_TURN_OFF_LED2();

    if (sixty_second_flag)
        batt_volt = ReadBatVoltage();
}

/* get radio ready...awakens in idle state */
SMPL_IOCTL( IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_AWAKE, 0);
IEN2 |= 0x01;                           // Enable RF interrupt
/* Set TID and designate which LED to toggle */
msg[3] = (batt_volt >> 12) & 0x0F;
msg[2] = (batt_volt >> 4) & 0xFF;
msg[1] = ++sTid;

batt_alarm = bat_volt_lo;
photo_alarm = ((alarm_flag << 1) & photo_sensor_alarm);
ion_alarm = alarm_flag & ion_sensor_alarm;

msg[0] = BV(3) | batt_alarm | photo_alarm | ion_alarm;
done = 0;
while (!done)
{  
    noAck = 0;

    /* Try sending message MISSES_IN_A_ROW times looking for ack */
    for (misses = 0; misses < MISSES_IN_A_ROW; ++misses)  
    {  
        if (SMPL_SUCCESS == (rc = SMPL_SendOpt(sLinkID1, msg, sizeof(msg),  
            SMPL_TXOPTION_ACKREQ)))
        {  
            /* Message acked. We're done. Toggle LED 2 to indicate ack received. */
            BSP_TOGGLE_LED2();
            break;
        }
        if (SMPL_NO_ACK == rc)
        {  
            /* Count ack failures. Could also fail because of CCA and
             * we don't want to scan in this case.
             */
            noAck++;
        }
    }
    if (MISSES_IN_A_ROW == noAck)
    {  
        /* Message not acked. Toggle LED 2. */
        BSP_TOGGLE_LED2();
    #ifdef FREQUENCY_AGILITY
        /* Assume we're on the wrong channel so look for channel by
        * using the Ping to initiate a scan when it gets no reply. With
        * a successful ping try sending the message again. Otherwise,
        * for any error we get we will wait until the next button
        * press to try again.
        */
        if (SMPL_SUCCESS != SMPL_Ping(sLinkID1))
        {  
            done = 1;
        }
    #else
        done = 1;
    #endif /* FREQUENCY_AGILITY */
    }
    else  
    {  
        /* Got the ack or we don't care. We're done. */
        done = 1;
    }
}
/* radio back to sleep */
SMPL_IOctl(IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_SLEEP, 0);
IEN2 &= ~0x01;  // Disable RF interrupt
BSP_TURN_OFF_LED1();
}
}

void Init_8051()
{
    /* Initialize Smoke Detector I/O pins to Outputs*/
P2DIR |= 0x01;
P0DIR |= 0xf8;

    /* Set the upper five bits of P0 port to Zero */
P0 &= 0x07;
    /* Set P2_0 Low - Battery Voltage Enable pin */
P2 &= 0xfe;

    return;
}

uint16_t ReadBatVoltage(void)
{
    uint16_t adc_result;

    /* Set system clock source to HS XOSC, with no pre-scaling.
     * Ref. [clk]=>[clk_xosc.c]*/
    SLEEP &= ~SLEEP_OSC_PD;
    while( !(SLEEP & SLEEP_XOSC_S) );
    CLKCON = (CLKCON & ~(CLKCON_CLKSPD | CLKCON_OSC)) | CLKSPD_DIV_1;
    while (CLKCON & CLKCON_OSC);
    SLEEP |= SLEEP_OSC_PD;

    /* Enable Battery Voltage Monitor */
P2 |= 0x01;

    /* NWK_DELAY(1);*/ /* Wait for output to settle */

    /* ADC configuration :*/
    /* - [ADCCON1.ST] triggered
     * - 12 bit resolution
     * - Single-ended
     * - Single-channel, due to only 1 pin is selected in the ADCCFG register
* - Reference voltage is VDD on AVDD pin

* Note: - [ADCCON1.ST] must always be written to 11
* The ADC result is represented in two's complement.
*/
// Configure P0_2 as an ADC analog input pin */
ADCCFG = 0x04;

// Set [ADCCON1.STSEL] according to ADC configuration */
ADCCON1 = (ADCCON1 & ~ADCCON1_STSEL) | STSEL_ST | BIT1 | BIT0;

// Set [ADCCON2.SREF/SDIV/SCH] according to ADC configuration */
ADCCON2 = ADCCON2_SREF_AVDD | ADCCON2_SDIV_512 | ADCCON2_SCH_AIN2;

/* ADC conversion:
* The ADC conversion is triggered by setting [ADCCON1.ST = 1].
* The CPU will then poll [ADCCON1.EOC] until the conversion is completed.
*/

/* Set [ADCCON1.ST] and await completion (ADCCON1.EOC = 1) */
ADCCON1 |= ADCCON1_ST | BIT1 | BIT0;

/* Approximate delay of 304 usec. */
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);

/* NWK_DELAY (1); */ /* Allow A/D Ref voltage to settle - delay 32usec */
while(!(ADCCON1 & ADCCON1_EOC));

/* Store the ADC result from the ADCH/L register to the adc_result variable.
* The 4 LSBs in ADCL will not contain valid data, and are masked out.
* adc_result = ADCL & 0xF0;
adcs_result |= (ADCH << 8);

/* Disable Battery Voltage Monitor */
P2 &= 0xfe;
batt_volt_lo = 0; /* Battery low voltage flag to be added to packet for AP */

/* Threshold (1.10 volts) was calculated as follows: round(1.10 volts * 2047/3 Volts A/D reference) */
/* Check Voltage against threshold */
if ((adc_result >> 4) < BAT_VOLT_TRSH)
{
batt_volt_lo = 0x04; /* Voltage too low. Alert AP */
/* Clock control. Configures the Timer tick speed setting, division by 128.
* This results in a Timer tick frequency of 203.125 kHz.
*/
// CLKCON = (CLKCON & ~CLKCON_TICKSPD) | (7 << 3);

/* Timer 4 channel 0 compare value. Sets the initial value which the Timer
* is to count down from.
*/
//T4CC0 = 0x0F;

/* Turn On Buzzer for a chirp (8ms ON) */
P0_7 = 1;
/* NWK_DELAY(8); */ /* BSP_SleepFor (POWER_MODE_1,
SLEEP_1_MS_RESOLUTION, 8); */
P0_7 = 0;

/* Timer 4 control. Configuration:
* - Prescaler divider value: 128.
* - Interrupts enabled.
* - Down mode.
* The Timer is also cleared and started.
*/
// T4CTL = T4CTL_DIV_128 | T4CTL_START | T4CTL_OVFIM | 
T4CTL_MODE_DOWN;

/* Enables global interrupts (IEN0.EA = 1) and interrupts from Timer 3
* (IEN1.T3IE = 1).
*/
// EA = 1;
//T4IE = 1;
return (adc_result);
}

uint8_t ReadPhotoSensor(void)
{
    uint8_t photo_sensor_alrm;
    uint16_t adc_result;
    static uint16_t photo_sensor_dark, photo_sensor_light;

    /* Set system clock source to HS XOSC, with no pre-scaling.
     * Ref. [clk]=>[clk_xosc.c]
     */
    SLEEP &= ~SLEEP_OSC_PD;
    while( !(SLEEP & SLEEP_XOSC_S) );
    CLKCON = (CLKCON & ~(CLKCON_CLKSPD | CLKCON_OSC)) | CLKSPD_DIV_1;
    while (CLKCON & CLKCON_OSC);
    SLEEP |= SLEEP_OSC_PD;

    /* Enable Photo Sensor OP AMP - Set P0_5 */
    P0 |= 0x20;

    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16);    /* Wait about 16 usec. for OP AMP power up */
    BSP_Delay(16); /* NWK_DELAY(1); */ /* Wait for OP AMP output to settle */

    /* ADC configuration :
     * - [ADCCON1.ST] triggered
     * - 12 bit resolution
     * - Single-ended
     * - Single-channel, due to only 1 pin is selected in the ADCCFG register
     * - Reference voltage is VDD on AVDD pin
     * Note: - [ADCCON1.ST] must always be written to 11
* The ADC result is represented in two's complement.
*/

/* Configure P0_1 as AIN1 (Photo Sensor input) for ADC */
ADCCFG = 0x02;

// Set [ADCCON1.STSEL] according to ADC configuration */
ADCCON1 = (ADCCON1 & ~ADCCON1_STSEL) | STSEL_ST | BIT1 | BIT0;

// Set [ADCCON2.SREF/SDIV/SCH] according to ADC configuration */
ADCCON2 = ADCCON2_SREF_AVDD | ADCCON2_SDIV_512 | ADCCON2_SCH_AIN1;

/* ADC conversion :
* The ADC conversion is triggered by setting [ADCCON1.ST = 1].
* The CPU will then poll [ADCCON1.EOC] until the conversion is completed.
*/

/* Set [ADCCON1.ST] and await completion (ADCCON1.EOC = 1) */
/* Start ADC conversion of DARK sensor */
ADCCON1 |= ADCCON1_ST | BIT1 | BIT0;

BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);  /* Aproximate delay of 320us */

/* NWK_DELAY (1); */ /* Allow A/D Ref voltage to settle - delay 32usec */
while( !(ADCCON1 & ADCCON1_EOC));

/* Store the ADC result from the ADCH/L register to the adc_result variable.*/
* The 4 LSBs in ADCL will not contain valid data, and are masked out.
adc_result = ADCL & 0xF0;
adc_result |= (ADCH << 8);
photo_sensor_dark = adc_result >> 4;  /* ADC set to 512 decimation; has 12-bits resolution */

/* Disable */
NWK_DELAY(1);  /* 1 ms delay for now */

/* Turn on Infrared Emmiter */
P0 |= 0x40;

BSP_Delay(16);
BSP_Delay(16);

/* NWK_DELAY(1); */  /* 1 ms delay for now -- needs to be optimized */

/* Start ADC conversion of LIGHT sensor */
ADCCON1 |= ADCCON1_ST | BIT1 | BIT0;

BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);
BSP_Delay(16);

/* Approximate delay of 320 us*/

/* NWK_DELAY (1); */  /* Allow A/D Ref voltage to settle */
while( !(ADCCON1 & ADCCON1_EOC));

/* Store the ADC result from the ADCH/L register to the adc_result variable. */
/* The 4 LSBs in ADCL will not contain valid data, and are masked out. */
adc_result = ADCL & 0xF0;
adc_result |= (ADCH << 8);
photo_sensor_light = adc_result >> 4; /* ADC set to 512 decimation; has 12-bits resolution */

/* Turn off Infrared Emitter and OP Amp */
P0 &= 0x9f;

photo_sensor_alrm = 0; /* clear alarm */
/* Assumption: photo_sensor_light >= photo_sensor_dark */
photo_sensor_alrm = (photo_sensor_light - photo_sensor_dark) > PHOTO_ALARM_THRES ? 0x02 : 0x00;

return (photo_sensor_alrm);
}

uint8_t ReadIonSensor(void)
{
    uint8_t ion_sensor_alrm;
    uint16_t adc_result;
    static uint16_t ion_sensor_result;

    /* Set system clock source to HS XOSC, with no pre-scaling.
     * Ref. [clk]=>[clk_xosc.c]
     */
    SLEEP &= ~SLEEP_OSC_PD;
    while( !(SLEEP & SLEEP_XOSC_S) );
    CLKCON = (CLKCON & ~(CLKCON_CLOCKSPD | CLKCON_OSC)) | CLKSPD_DIV_1;
    while (CLKCON & CLKCON_OSC);
    SLEEP |= SLEEP_OSC_PD;

    /* Enable Ionization Sensor Bias */
P0 |= 0x10;

    /* NWK_DELAY(1); /* Wait for OP AMP output to settle */

    /* ADC configuration:
     * - [ADCCON1.ST] triggered
     * - 12 bit resolution
     * - Single-ended
     * - Single-channel, due to only 1 pin is selected in the ADCCFG register
     * - Reference voltage is VDD on AVDD pin
     * Note: - [ADCCON1.ST] must always be written to 11
     *
     * The ADC result is represented in two's complement.
     */
    /* Configure P0_0 as AIN0 (Ionization Sensor input) for ADC */
    ADCCFG = 0x01;
// Set [ADCCON1.STSEL] according to ADC configuration */
ADCCON1 = (ADCCON1 & ~ADCCON1_STSEL) | STSEL_ST | BIT1 | BIT0;

// Set [ADCCON2.SREF/SDIV/SCH] according to ADC configuration */
ADCCON2 = ADCCON2_SREF_AVDD | ADCCON2_SDIV_512 | ADCCON2_SCH_AIN0;

/* ADC conversion :
* The ADC conversion is triggered by setting [ADCCON1.ST = 1].
* The CPU will then poll [ADCCON1.EOC] until the conversion is completed.
*/

/* Set [ADCCON1.ST] and await completion (ADCCON1.EOC = 1) */
/* Start ADC conversion of Ionization Sensor */
ADCCON1 |= ADCCON1_ST | BIT1 | BIT0;

NWK_DELAY (1); /* Allow A/D Ref voltage to settle - delay 32usec */
while( !(ADCCON1 & ADCCON1_EOC));

/* Store the ADC result from the ADCH/L register to the adc_result variable.
* The 4 LSBs in ADCL will not contain valid data, and are masked out.
*/
adc_result = ADCL & 0xF0;
adc_result |= (ADCH << 8);
ion_sensor_result = adc_result >> 4; /* ADC set to 512 decimation; has 12-bits resolution */

/* Turn off Ionization Sensor bias */
P0 &= 0xef;

ion_sensor_alrm = 0;
ion_sensor_alrm = (ion_sensor_result > ION_ALARM_THRES) ? 0x01 : 0x00;

return (ion_sensor_alrm);
}

void BuzzerSetup(void)
{

}
T1IE = 1;

// Enable global interrupt by setting the [IEN0.EA=1]
EA = 1;

// Set compare register of channel 0 to 50781 (to get a 500 msec on/500 msec off buzzer sound)
T1CC0L = 0x5D;
T1CC0H = 0xC6;

// Set prescalar divider value to 128 to get a tickspeed of 101.56 kHz (13 MHz/128) and
// keep Timer 1 off. It will be enabled in the main thread when there is an alarm.
T1CTL |= 0x0c;

#pragma vector = T1_VECTOR
__interrupt void timer1_ISR(void)
{
    BSP_DISABLE_INTERRUPTS();
    P0 ^= 0x80;              /* Toggle Buzzer Enable pin P0_7 at a rate of 1 sec (500 ms on/500ms off) */
    T1CTL = (T1CTL & 0xEF);  /* Clear T1 OVFIF interrupt flag */
    //T1CTL = (~T1CTL_OVFIF & 0xF0) | (T1CTL & 0x0F);
    BSP_ENABLE_INTERRUPTS();
}

void delay_msec(uint8_t msec)
{
    uint8_t tickCount;
    tickCount = (msec * BSP_CLOCK_MHZ); /* Calculate the tick count. */
    T4CTL &= ~(BV(4)); /* Stop the timer. */
    T4CTL &= ~(BV(3)); /* Clear Interrupt mask flag. */
    TIMIF &= ~(BV(3)); /* Clear the interrupt flag. */
    T4CTL |= BV(2); /* Reset the count value. */
    T4CC0 = tickCount; /* Set the compare value. */
    T4CTL = (T4CTL & (~0xE3)) | 0xE1; /* Set down mode and prescaler /128 */
    T4CTL |= BV(4); /* Start the timer. */

    while( !(TIMIF & BV(3)) ); /* Wait till interrupt flag is set. i.e. count is reached. */
    T4CTL &= ~(BV(4)); /* Stop the timer. */
    TIMIF &= ~(BV(3)); /* Clear the interrupt flag. */
}
The following code listing implements the main flowchart of Figure 9 above.

`AD Wireless AP Main Program` /* AD Wireless AP Main Program */

`__________________________________________________________________________`

`******************************************************************************
** Copyright 2007-2009 Texas Instruments Incorporated. All rights reserved.**
******************************************************************************

IMPORTANT: Your use of this Software is limited to those specific rights granted under the terms of a software license agreement between the user who downloaded the software, his/her employer (which must be your employer) and Texas Instruments Incorporated (the "License"). You may not use this Software unless you agree to abide by the terms of the License. The License limits your use, and you acknowledge, that the Software may not be modified, copied or distributed unless embedded on a Texas Instruments microcontroller or used solely and exclusively in conjunction with a Texas Instruments radio frequency transceiver, which is integrated into your product. Other than for the foregoing purpose, you may not use, reproduce, copy, prepare derivative works of, modify, distribute, perform, display or sell this Software and/or its documentation for any purpose.

YOU FURTHER ACKNOWLEDGE AND AGREE THAT THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TEXAS INSTRUMENTS OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.

Should you have any questions regarding your right to use this Software, contact Texas Instruments Incorporated at www.TI.com.
#include <string.h>
#include "bsp.h"
#include "mrfi.h"
#include "bsp_leds.h"
#include "bsp_buttons.h"
#include "nwk_types.h"
#include "nwk_api.h"
#include "nwk_frame.h"
#include "nwk.h"

#include "app_remap_led.h"
#include "project_config.h"

#ifndef APP_AUTO_ACK
#error ERROR: Must define the macro APP_AUTO_ACK for this application.
#endif

void toggleLED(uint8_t);

/**************************** COMMENTS ON ASYNC LISTEN
APPLICATION ***********************

Summary:
This AP build includes implementation of an unknown number of end device peers in
addition to AP functionality. In this scenario all End Devices establish a link to
the AP and only to the AP. The AP acts as a data hub. All End Device peers are on
the AP and not on other distinct ED platforms.

There is still a limit to the number of peers supported on the AP that is defined
by the macro NUM_CONNECTIONS. The AP will support NUM_CONNECTIONS or
fewer peers
but the exact number does not need to be known at build time.

In this special but common scenario SimpliciTI restricts each End Device object to a
single connection to the AP. If multiple logical connections are required these must
be accommodated by supporting contexts in the application payload itself.

Solution overview:
When a new peer connection is required the AP main loop must be notified. In essence
the main loop polls a semaphore to know whether to begin listening for a peer Link
request from a new End Device. There are two solutions: automatic notification and
external notification. The only difference between the automatic notification
solution and the external notification solution is how the listen semaphore is
set. In the external notification solution the semaphore is set by the user when the
AP is stimulated for example by a button press or a commend over a serial link. In the
The automatic scheme the notification is accomplished as a side effect of a new End Device joining.

The Rx callback must be implemented. When the callback is invoked with a non-zero Link ID the handler could set a semaphore that alerts the main work loop that a SMPL_Receive() can be executed successfully on that Link ID.

If the callback conveys an argument (LinkID) of 0 then a new device has joined the network. A SMPL_LinkListen() should be executed.

Whether the joining device supports ED objects is indirectly inferred on the joining device from the setting of the NUM_CONNECTIONS macro. The value of this macro should be non-zero only if ED objects exist on the device. This macro is always non-zero for ED-only devices. But Range Extenders may or may not support ED objects. The macro should be set to 0 for REs that do not also support ED objects. This prevents the Access Point from reserving resources for a joining device that does not support any End Device Objects and it prevents the AP from executing a SMPL_LinkListen(). The Access Point will not ever see a Link frame if the joining device does not support any connections.

Each joining device must execute a SMPL_Link() after receiving the join reply from the Access Point. The Access Point will be listening.

END COMMENTS ON ASYNC LISTEN APPLICATION

/* reserve space for the maximum possible peer Link IDs */
static linkID_t sLID[NUM_CONNECTIONS] = {0};
static uint8_t  sNumCurrentPeers = 0;

/* callback handler */
static uint8_t sCB(linkID_t);

/* received message handler */
static void processMessage(linkID_t, uint8_t *, uint8_t);

static void BuzzerSetup(void);
extern void delay_msec(uint8_t msec);
__interrupt void timer1_ISR(void);

/* Frequency Agility helper functions */
static void checkChangeChannel(void);
static void changeChannel(void);

/* work loop semaphores */
static volatile uint8_t sPeerFrameSem = 0;
static volatile uint8_t sJoinSem = 0;

volatile uint8_t alarm_flag;
static uint8_t batt_voltH;
static uint8_t batt_voltL;
//@static uint16_t batt_voltage;

#ifdef FREQUENCY_AGILITY
/* ************** BEGIN interference detection support */

#define INTERFERNCE_THRESHOLD_DBM (-70)
#define SSIZE 25
#define IN_A_ROW 3
static int8_t sSample[SSIZE];
static uint8_t sChannel = 0;
#endif /* FREQUENCY_AGILITY */

/* blink LEDs when channel changes... */
static volatile uint8_t sBlinky = 0;

/* ************** END interference detection support */

#define SPIN_ABOUT_A_QUARTER_SECOND NWK_DELAY(250)

#define VOLTAGE_RATIO 3/2047
/* Strings */
//@const rom char *const rom hello = "Hello, World!";
//@const rom char *const rom second_line = "I made it!";
const char *const hello = "AD_WsSmk";
const char *const second_line = "DetAlarm";
const char *const hello1 = "Bat Volt";

void main (void)
{
    bspiState_t intState;

    /* memset(sSample, 0x0, sizeof(sSample)); */
    P0DIR |= 0x80;                // Configure P0_7 as output to drive the alarm buzzer */
P1DIR |= 0x10; // Configure P1_4 as output to drive phone dialer enabling relay */
P0_7 = 1; // Make sure Buzzer(P0_7) is disabled at startup */
P1_4 = 0; // Auto Dialer phone call trigger is disabled at startup */

SPI_74HC595_init(); //initialize SPI to work with 74HC595
LCD_init();
LCD_send_string( hello );
LCD_second_row();
LCD_send_string( second_line );

BSP_Init();

BuzzerSetup();

/* If an on-the-fly device address is generated it must be done before the * call to SMPL_Init(). If the address is set here the ROM value will not * be used. If SMPL_Init() runs before this IOCTL is used the IOCTL call * will not take effect. One shot only. The IOCTL call below is conformal. */
#endif /* I_WANT_TO_CHANGE_DEFAULT_ROM_DEVICE_ADDRESS_PSEUDO_CODE */

SMPL_Init(sCB);

/* green and red LEDs on solid to indicate waiting for a Join. */
if (!BSP_LED2_IS_ON())
{
    toggleLED(2);
}
if (!BSP_LED1_IS_ON())
{
    toggleLED(1);
}

/* main work loop */
while (1)
{

}
/* Wait for the Join semaphore to be set by the receipt of a Join frame from a * device that supports an End Device. */

* An external method could be used as well. A button press could be connected * to an ISR and the ISR could set a semaphore that is checked by a function * call here, or a command shell running in support of a serial connection * could set a semaphore that is checked by a function call. */

if (sJoinSem && (sNumCurrentPeers < NUM_CONNECTIONS))
{
    /* listen for a new connection */
    while (1)
    {
        if (SMPL_SUCCESS == SMPL_LinkListen(&sLID[sNumCurrentPeers]))
        {
            break;
        }
        /* Implement fail-to-link policy here. otherwise, listen again. */
    }

    sNumCurrentPeers++;

    BSP_ENTER_CRITICAL_SECTION(intState);
    sJoinSem--;
    BSP_EXIT_CRITICAL_SECTION(intState);
}

/* Have we received a frame on one of the ED connections? */
* No critical section -- it doesn't really matter much if we miss a poll */
if (sPeerFrameSem)
{
    uint8_t msg[MAX_APP_PAYLOAD], len, i;

    /* process all frames waiting */
    for (i=0; i<sNumCurrentPeers; ++i)
    {
        if (SMPL_SUCCESS == SMPL_Receive(sLID[i], msg, &len))
        {
            processMessage(sLID[i], msg, len);

            BSP_ENTER_CRITICAL_SECTION(intState);
            sPeerFrameSem--;
            BSP_EXIT_CRITICAL_SECTION(intState);
        }
    }
void toggleLED(uint8_t which)
{
    if (1 == which || 8 == which)
    {
        BSP_TOGGLE_LED1();
    }
    else if (2 == which)
    {
        BSP_TOGGLE_LED2();
    }

    return;
}

/* Runs in ISR context. Reading the frame should be done in the */
/* application thread not in the ISR thread. */
static uint8_t sCB(linkID_t lid)
{
    if (lid)
{  
sPeerFrameSem++;  
sBlinky = 0;  
}
else  
{
  sJoinSem++;  
}

/* leave frame to be read by application. */
return 0;
}

static void processMessage(linkID_t lid, uint8_t *msg, uint8_t len)
{
  uint8_t alarm_stat;
  //uint16_t batt_voltage;

  /* do something useful */
  if (len)  
  {  
    toggleLED(*msg);
    alarm_stat = *msg & 0xF7;
    batt_voltL = *(msg + 2);
    batt_voltH = *(msg + 3);
    //batt_voltage = (float) (batt_voltH << 8 | batt_voltL);
    //batt_voltage = (batt_voltage * 3.0)/ 2047.0;

    LCD_Home();
    NWK_DELAY(1);
    LCD_send_string( hello1 );
    NWK_DELAY(1);
    LCD_second_row();
    NWK_DELAY(1);
    LCD_send_hexbyte(batt_voltH);
    NWK_DELAY(1);
    LCD_send_hexbyte(batt_voltL);

    if (alarm_stat) {
      P1_4 = 1;        /* Activate phone dialer to make emergency phone call */
      T1CTL |= 0x0E;   /* Enable Buzzer upon an alarm */
    }
  } else {
    P1_4 = 0;
    P0_7 = 1;
T1CTL &= 0xFC; /* Disable Buzzer when alarm goes away */
}
return;
}

static void changeChannel(void)
{
#ifdef FREQUENCY_AGILITY
    freqEntry_t freq;
    if (++sChannel >= NWK_FREQ_TBL_SIZE)
    {
        sChannel = 0;
    }
    freq.logicalChan = sChannel;
    SMPL_Ioctl(IOCTL_OBJ_FREQ, IOCTL_ACT_SET, &freq);
    BSP_TURN_OFF_LED1();
    BSP_TURN_OFF_LED2();
    sBlinky = 1;
#endif
    return;
}

/** implement auto-channel-change policy here... **/ static void checkChangeChannel(void)
{
#ifdef FREQUENCY_AGILITY
    int8_t dbm, inARow = 0;
    uint8_t i;
    memset(sSample, 0x0, SSIZE);
    for (i=0; i<SSIZE; ++i)
    {
        /* quit if we need to service an app frame */
        if (sPeerFrameSem || sJoinSem)
        {
            return;
        }
        NK_DELAY(1);
        SMPL_Ioctl(IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_RSSI, (void *)&dbm);
        sSample[i] = dbm;
        if (dbm > INTERFERNCE_THRESHOLD_DBM)
        {
            /* implement auto-channel-change policy here... */
            static void changeChannel(void)
            {
                freqEntry_t freq;
                if (++sChannel >= NWK_FREQ_TBL_SIZE)
                {
                    sChannel = 0;
                }
                freq.logicalChan = sChannel;
                SMPL_Ioctl(IOCTL_OBJ_FREQ, IOCTL_ACT_SET, &freq);
                BSP_TURN_OFF_LED1();
                BSP_TURN_OFF_LED2();
                sBlinky = 1;
            }
            #endif
        }
    }

    return;
}

}
if (++inARow == IN_A_ROW)
{
    changeChannel();
    break;
}
}
else
{
    inARow = 0;
}
}
#endif
return;
}

void delay_msec(uint8_t msec)
{
    uint8_t tickCount;

tickCount = (msec * BSP_CLOCK_MHZ);  /* Calculate the tick count. */
T4CTL &= ~(BV(4));                  /* Stop the timer. */
T4CTL &= ~(BV(3));                  /* Clear Interrupt mask flag. */
TIMIF &= ~(BV(3));                  /* Clear the interrupt flag. */
T4CTL |= BV(2);                     /* Reset the count value. */
T4CC0  = tickCount;                 /* Set the compare value. */
T4CTL  = (T4CTL & ~(0xE3)) | 0xE1;  /* Set down mode and prescaler 128 */
T4CTL |= BV(4);                     /* Start the timer. */

while( !(TIMIF & BV(3)) );          /* Wait till interrupt flag is set, i.e. count is reached. */
    T4CTL &= ~(BV(4));         /* Stop the timer. */
    TIMIF &= ~(BV(3));        /* Clear the interrupt flag. */
void BuzzerSetup(void)
{
    /***************************************************************************
    /* Setup Timer 1 and enable its interrupt.                                */
    /* Timer 1 will generate a signal with a period of 1 second and a 50% duty */
    /* cycle that will drive the AP buzzer when there is an Alarm.           */
    /***************************************************************************
    
    OVFIM = 1;                    // Enable Timer 1 overflow interrupt mask
    // Enable Timer 1 interrupt so that it is ready to drive the Buzzer when there ia an alarm.
    T1IE = 1;
    // Enable global interrupt by setting the [IEN0.EA=1]
    EA = 1;
    // Set compare register of channel 0 to 50781 ( to get a 500 msec on/ 500 msec off
    buzzer sound)
    T1CC0L = 0x5D;
    T1CC0H = 0xC6;

    // Set prescalar divider value to 128 to get a tickspeed of 101.56 kHz (13 MHz/128) and
    // keep Timer 1 off. It will be enabled in the main thread when there is an alarm.
    T1CTL |= 0x0c;
}

#pragma vector = T1_VECTOR
__interrupt void timer1_ISR(void)
{
    BSP_DISABLE_INTERRUPTS();
    P0 ^= 0x80;              /* Toggle Buzzer Enable pin P0_7  at a rate of 1 sec (500 ms on/ 500ms off) */
    T1CTL = (T1CTL & 0xEF);  /* Clear T1 OVFIF interrupt flag */
    BSP_ENABLE_INTERRUPTS();
}