Wi-Fi Smart Home Control

Controlling home appliances from your Smartphone

Project Report

Submitted by:
Iyad Kuwatly-
Pavankumar Bhujannavar-
Shivesh Kumar Sharma-

Course ID: ECE5620
Group 3
Fall 2013

Prof. Syed M. Mahmud
Department of Electrical and Computer Engineering
Wayne State University, Detroit MI

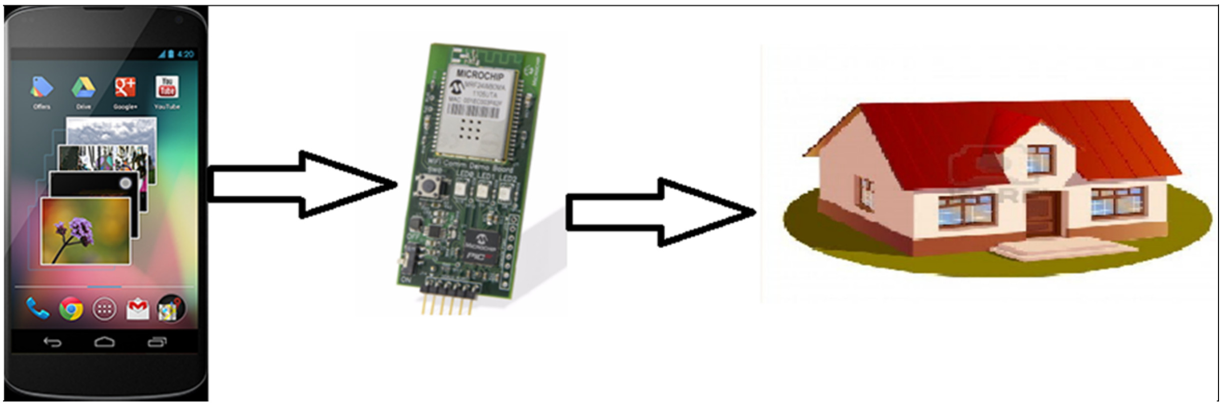
INDEX:

ABSTRACT:

Wireless Sensor Networks are providing tremendous benefit for a number of industries. A smart home control system can be designed based on presence information provided by all devices or objects in the home, and we can act depending on it. We explain its components and features that facilitate the daily living of person's own home.

Smart Home technology which was started for more than a decade with a concept to introduce the concept of Networking devices and equipment's in the house. According to the Smart Homes Association the best definition of smart home technology is the integration of technology and services through home networking for a better quality of living. Many tools that are used in computer systems or in mobile phones like android (applied in this project) can also be integrated in Smart Home Systems.

A sample house environment that is one branch of the Smart home is addressed in this paper. The system is based on the PIC microcontroller software and can act as an automatic control in the home. The system can monitor the temperature, lighting different LED's located indifferent part of house. The system also has Wi-Fi connection to the PIC microcontroller to control the house equipment's from distant parts of house. This project report presents the hardware implementation of a smart home control system using PIC microcontroller that is wirelessly connected to an android phone(using android programing an application has been generated which can receive as well as can send a signal to PIC microcontroller to which different appliances are connected in house). Such a system belongs to a domain usually named smart house systems. The approach combines hardware and software technologies. Using smart Wi-Fi home controller home appliances can be controlled by your phone by creating a user interface in your mobile phone which will be connected to the PIC microcontroller through Wi-Fi connection and all other home appliances will be connected to PIC.



Introduction:

The basic idea of Smart home system is to employ sensors and control systems to monitor and accordingly adjust the various mechanisms that provide heat, ventilation, lighting, and other services. By more closely tuning the mechanical systems to the dweller's needs, the automated "intelligent" home can provide a safer, more comfortable, and more economical living. For example, the electronic controller of an automated home can determine when the person has gone to bed and turn off the lights and check the temperature of home; it can monitor all of these electric home devices using mobile phone.

- Control includes the following devices:
  - House Lights (applied in prototype)
  - Temperature Sensor (applied in prototype)
  - Smoke (CO2) Detectors
  - Motion Sensors
  - Water Sensors

What do we expect from Smart Homes?

- Safety
- comfort and Entertainment
- Efficiency
- Connectivity

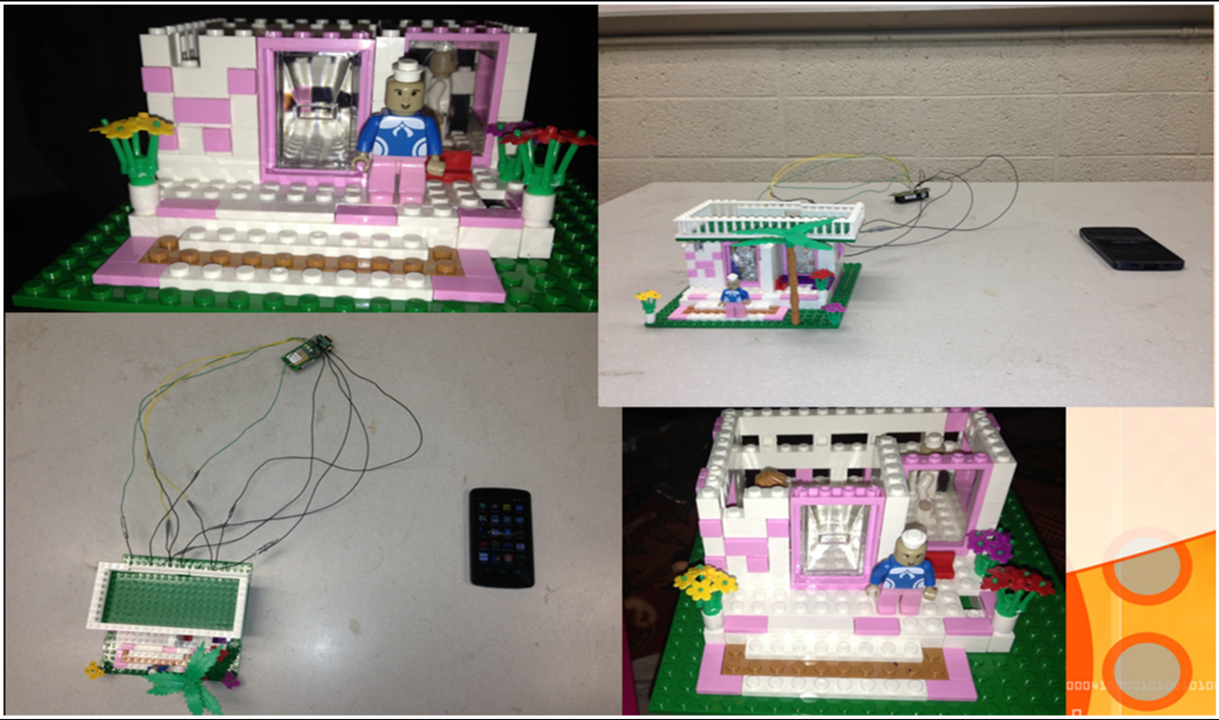
Using Smart Home energy can be saved:

1. Lighting control is not about reducing light. It offers the correct light when and where required, while reducing wastage.
2. Enjoy optimal scenes by recalling the optimized pre-sets every time for sustainable energy-saving.
3. Using smart home system you can minimize wastage of energy without affecting usage with occupancy detection.

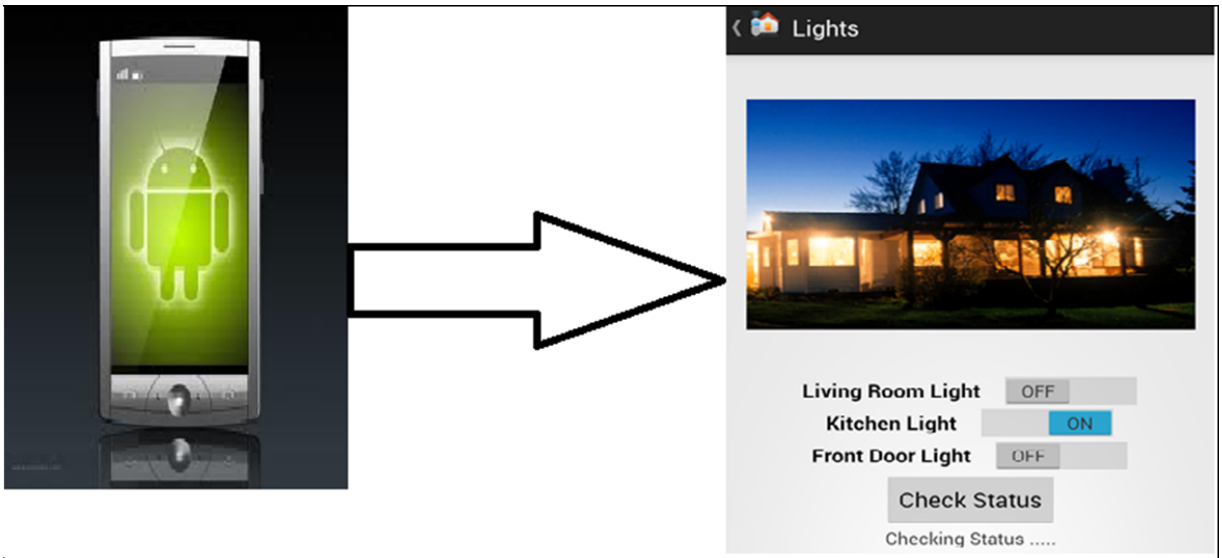
The limitless possibilities that open up when you are fully connected:

1. Use your mobile as a window to your house.
2. Control all your devices from your mobile.
3. Receive an email when the system detects an unexpected event.
4. Combine remote video monitoring with your security system.
5. Use your mobile to turn ON and OFF lights in your house even when you are not there.
6. By using your mobile phone you can sense the temperature in your house before entering in it.

Prototype Model:



Base Prototype idea:



In above figure it is shown that you can control different lights in home by using mobile phone.

Background Materials, Design and Technical details:

Design and Technical details can be split into the following:

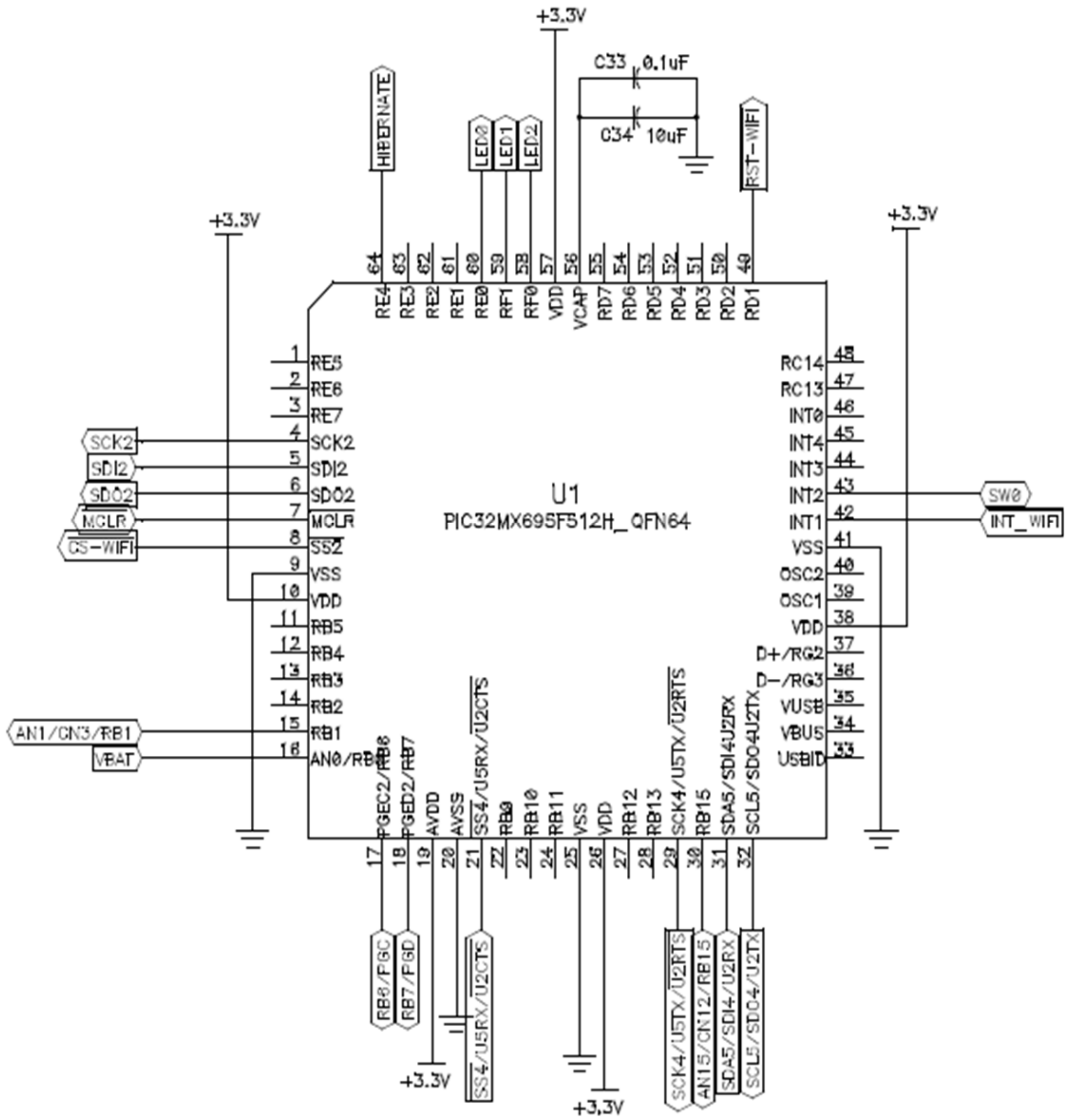
1. Understanding of PIC 32 Family Microcontroller- PIC Model #: PIC32MX695F512H
2. Sensors
3. Actuators
4. Wireless RF Transceiver- Microchip MRF24WB0MA- RF Transceiver
5. Wi-Fi Communication
6. Android Application

PIC32 Microcontroller- PIC Model #: PIC32MX695F512H

PIC32 family of Microcontroller was chosen to drive the main idea of Project and Wi-Fi Communication

PIC32 features:

1. 5 Stage pipeline, Harvard architecture
2. Operating Speed of 80MHz
3. Single cycle multiply and hardware divide unit
    a. 32 x 32-bit Core Registers
    b. 32 x 32-bit Shadow Registers
4. Fast context switch and interrupt response
5. 512 K Program Memory
6. 128K RAM (can execute from RAM)
7. Flash prefetch module with 256 Byte cache
8. Lock instructions or data in cache for fast access via Prefetch
9. Programmable interrupt controller- interrupts can be prioritised
10. Fast and Accurate 16 channel 10-bit ADC
11. Max Resolution 10
    a. 2 Inbuilt comparators
    b. 2 Internal oscillators (8MHz & 31 KHz)
12. Hardware RTCC (Real-Time Clock and Calendar with Alarms)
13. Pin compatible with 16-bit PIC MCUs
14. Serial Communication Modules allow flexible UART (Universal Asynchronous Receiver/Transmitter- used for serial communication over a computer or peripheral device)/SPI (Serial Peripheral Interface- Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed to connect with individual slave select lines.)/I2C (Inter-Integrated Circuit configuration - used for attaching low-speed peripherals to a motherboard, embedded system, cell phone or other electronic device)



PIC 32 Pin Diagram

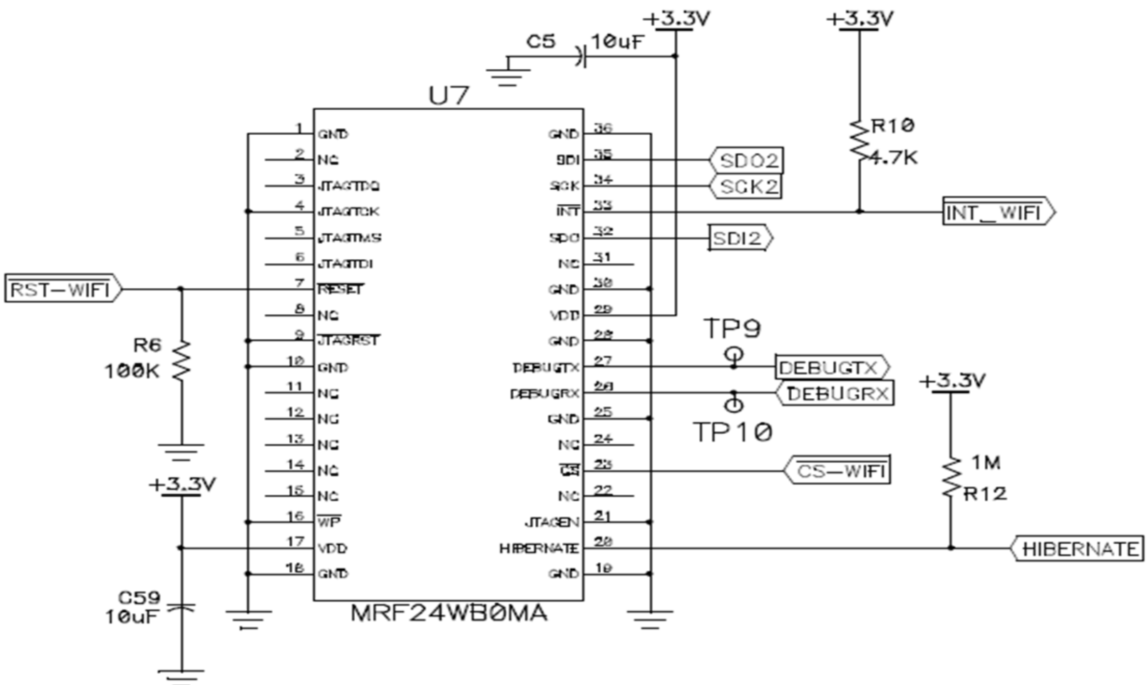
Microchip MRF24WB0MA- RF Transceiver

MRF24WB0MA features:

1. IEEE Std. 802.11-compliant RF Transceiver
2. Serialized unique MAC address
3. Data Rate: 1 and 2Mbps (MRF24WB0MA and MRF24WB0MB resp)
4. Small size: 21mm x 31mm 36-pin Surface Mount Module
5. Integrated PCB antenna (MRF24WB0MA)
6. External antenna option (MRF24WB0MB)
7. Easy to integrate with other modules
8. Radio regulation certification for United States (FCC), Canada (IC), Europe (ETSI) and Japan (ARIB)
9. Designed for use with Microchip microcontroller families (PIC18, PIC24, dsPIC33, and PIC32)

Operational Features

1. Single operating voltage: 2.7V-3.6V (3.3V typical)
2. Temperature Range: 0°C to +70°C Commercial
3. Band 2.400-2.483.5GHz operation
4. Data Rate - 1000 kbps



Pin diagram of MRF24WB0MA RF transceiver

Wi-Fi Functionality:

In order to communicate over long distances without running wires, we came up with a convenient way of communicating with sensors. The different I/O devices are controlled using TCP/IP over the IEEE 802.11g standard protocol. Data being gathered from sensors, such as temperature sensors, light sensors, and laser tripwire sensors, is being processed on PIC32 Micro-controller and then broadcast with an attached to a server using the TCP/IP protocol. PIC32 has a statically assigned IP address that corresponds to an individual room in the house. Each time a request is made to that IP address, an HTML page is returned with implemented functionality. One of the perks of using HTML is that, data can be viewed from all of the sensors in one location, and input/output devices such as power strip plugs can be remotely controlled.

Why we choose wireless signal transmission?

When someone gets home and wants to enable light, fan, music at home, he may use his SMART Phone to control his household appliances. However, the transmission distance of the signal is short. So we decide to place a central control unit in the centre of the room and send digital signal wirelessly to where other microcontroller is. Once the receiver gets the signal, it will operate on/off/dimmer function which has been programmed in it previously.

Microchip's Wi-Fi Communication Board:

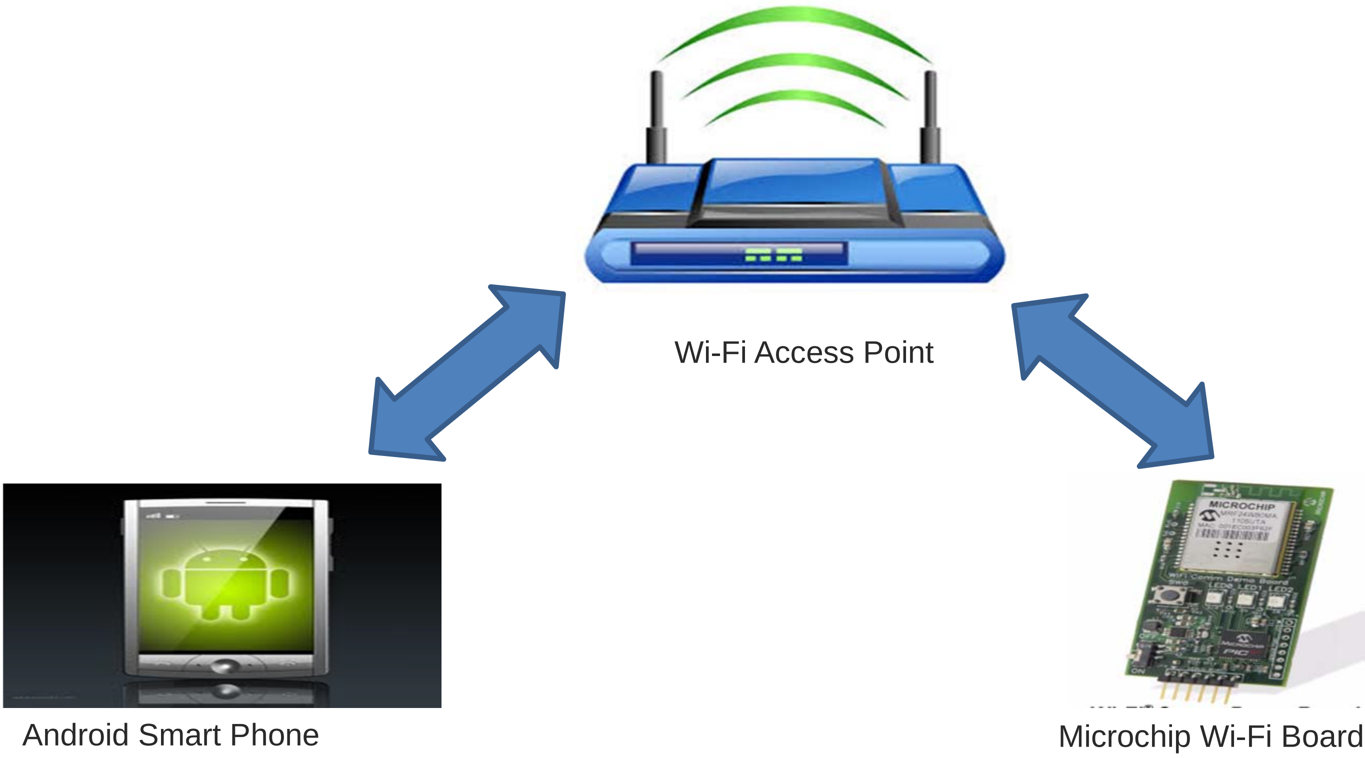
Microchip's DV102411 Wi-Fi Comm Demo Board combines Microchip's 32-bit PIC32 microcontroller family with its low-power MRF24WB0MA certified, IEEE 802.11 and embedded Wi-Fi radio transceiver module.

Microchip also provides a full-featured TCP/IP stack. This board is compact and cost-effective demo board is designed to be easy to integrate with existing embedded designs to evaluate Wi-Fi connectivity and 32-bit performance with minimal effort.

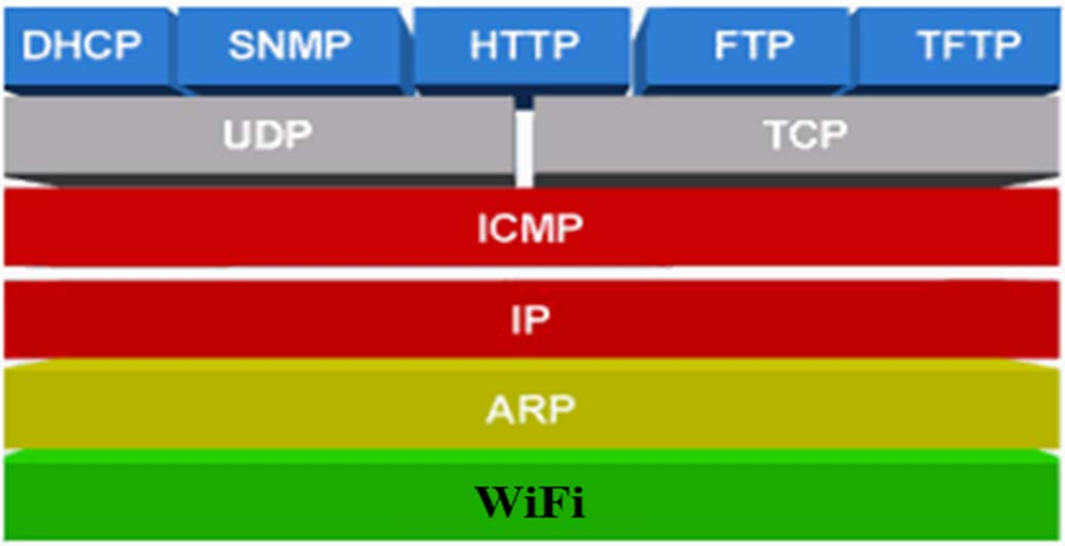
Features of Board:

Wi-Fi solution compatible with IEEE 802.11 b/g Access Points
Supports Infrastructure and Ad hoc networks
Can be powered by 2 AAA batteries
Supports security protocols (WEP, WPA and WPA2)
The board has sensor I/O interface enabling application

Basic Communication Analogy:



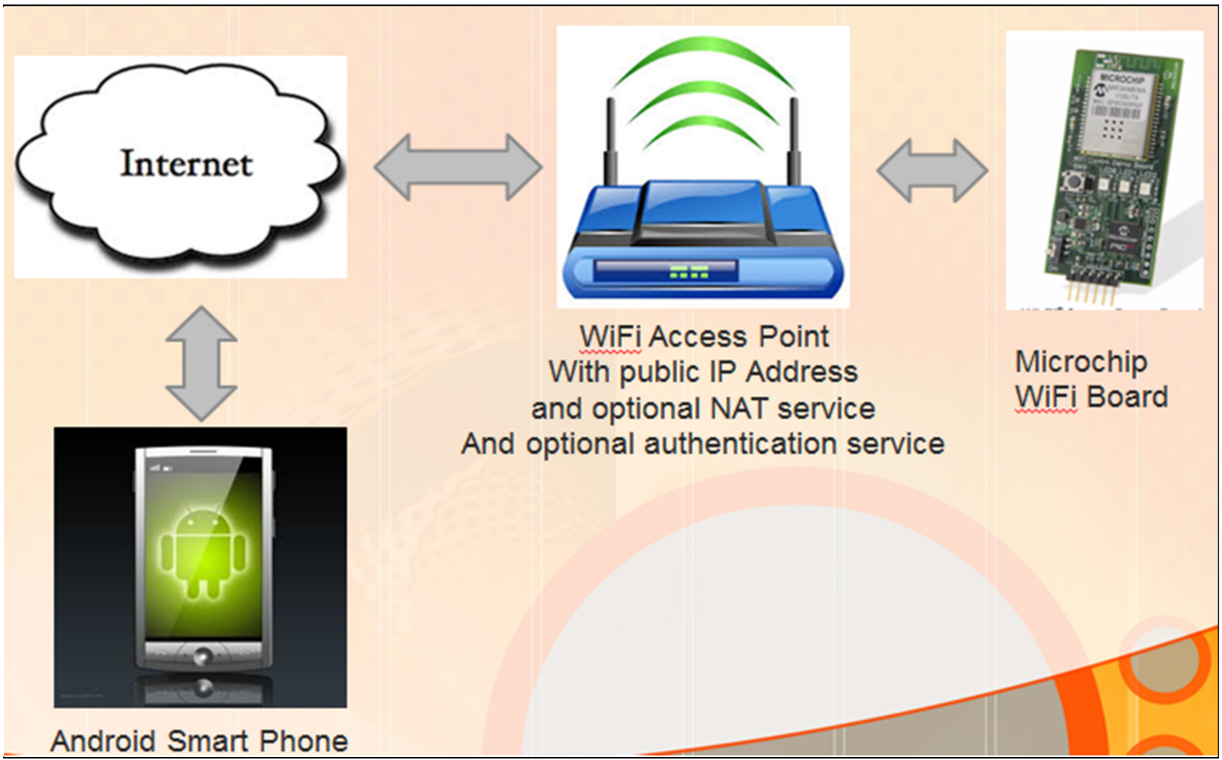
The project makes use of Wi-Fi IEEE 802.11 standard to insure the physical layer communications between the PIC32 and android app. The following diagram illustrates the IP implementation used in PIC 32



The Pic32 IP stacked was configured to use TCP (Transmission Control Protocol) in the transport layer to insure reliable communications and in the application layer, we made use of HTTP (The hypertext transfer protocol).

The Wi-Fi module supports both Adhoc mode and infrastructure mode. Because android did not support Adhoc mode, we had to use access point to be insure communication between the PIC32 with Wi-Fi module and android phone.

To be able to control the house beyond the range limited by the Wi-Fi access point which can go up to 300 m using infrastructure mode., the following diagram illustrates a possible approach.



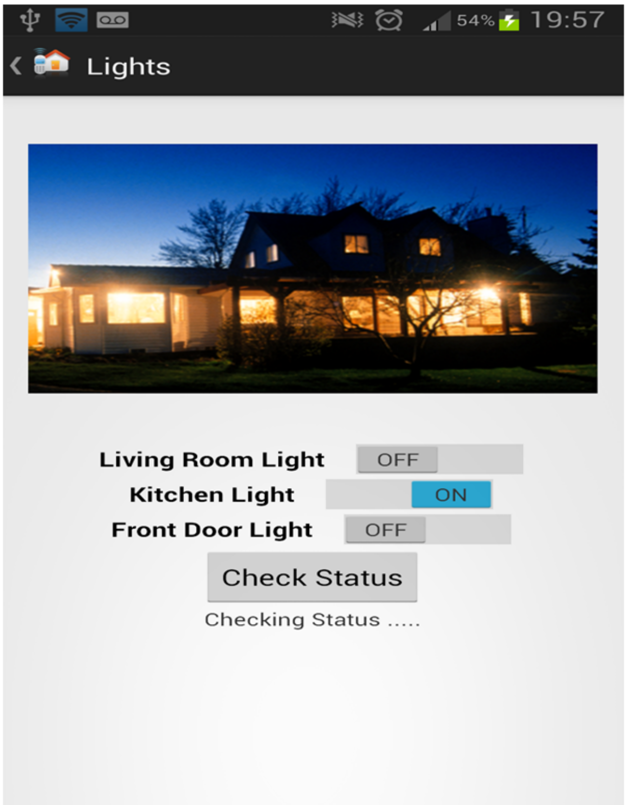
The figure proposes that the android phone is to be connected to the internet using any available technology in the phone which includes: 4th Generation mobile access such as LTE (Long Term Evolution) or earlier generations mobile access such as EDGE or GPRS or simply though Wi-Fi network located in a café or university campus for example. Now that the phone is connected to the internet, we configure out Wi-Fi at home to access the internet too. The home Wi-Fi can be configured with a public IP address which is provided by the ISP (Internet service provider) with a little extra charge on the monthly internet subscription.

The result is our home Wi-Fi is now accessible from anywhere; on the other hand an very important challenge would rise. Securing access to the Wi-Fi is now very important. Solutions to this issue include: 1- adding a layer of security where the user has to go through authentication process. 2- Encrypting communication between android and pic 32 for example using https) Hypertext Transfer Protocol Secure) instead of the normal http

Android Application:

The android app is the user interface that provides home control. The app consists of three activities (user screens). 1- Light control 2- Temperature Readings 3- IP Configuration

The first activity is used for light control. The following figure illustrates its design:



It consists mainly of three switches and one button. To respond the user's interactions with buttons and switch, listeners were implemented in Java as follows:

```
s1.setOnCheckedChangeListener((buttonView, isChecked) -> {
        if (localCall)
        {
            final TextView mTextView = (TextView) findViewById(R.id.txtStatus);
            mTextView.setText("Toggling light 1....");

            toggleLight(IPAddress + "leds.cgi?led=0");

        }
});
```

When the user click on the switch, localCall global variable is check to differentiate if the call was initiated by the user or it is a result of program call back. The reason to do that is when the program checks for any updates in the status of the light, it triggers the switch and this change in the switch status again call the CheckedChangeListener. This way we avoid recursive calls of the listener function.

Next in the code we set a comment “Toggling light 1 …” for the user. The toggleLight function is passed the IP address of the microchip together with the action to be performed. The action being toggle led 0.

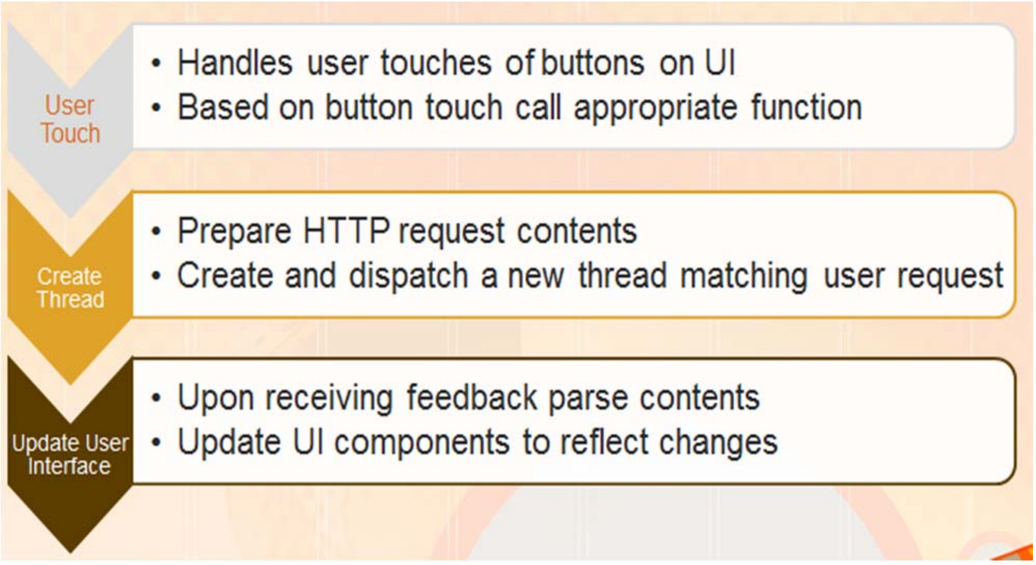
The following figure shows toggleLight function which is run as a separate thread that is dispatched from the user activity. This way the user would have the convenience of toggling multiple lights without having to wait for the response from the PIC controller.

```java
private void toggleLight(final String path) {
    (Thread) run() -> {

            URL u = null;
            try {
                u = new URL(path);
                HttpURLConnection c = (HttpURLConnection) u.openConnection();
                c.setRequestMethod("GET");
                c.connect();
                InputStream in = c.getInputStream();
                final ByteArrayOutputStream bo = new ByteArrayOutputStream();
                byte[] buffer = new byte[1024];
                in.read(buffer); // Read from Buffer.
                bo.write(buffer); // Write Into Buffer.
                runOnUiThread(() -> {
                        try {
                            bo.close();
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                });

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (ProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }

    }.start();
}
```

The code first open an http connection and use GET method of the HTTP protocol to be able collect the feedback from the microchip after it receives the request. The connection is established on port 80 the default http port. Multiple catch blocks were implemented to assist us in the debugging process the including: failure in protocol stack implementation or malformed reply.

It important to mention that the http protocol uses TCP (Transmission control Protocol) in order to establish the connection. The TCP offer reliable data communication and flow control. This was particularly important for us, because the physical medium is wireless which is very prone to errors. The IP stack implementation on both the android phone and the Pic 32 microchip made this possible and guaranteed retransmission of any packets in error or lost packets automatically and seamlessly as part of the TCP implementation.

The following flow chart illustrates the steps implemented to handle user interaction.



The last step illustrates the handling of feedback received from the microchip PIC 32 to update user interface components. To begin the discussion, it is important to look at a sample XML (Extensible Markup Language) file containing the status feedback.

```xml
<?xml version="1.0" encoding="utf-8"?>

 <response>
    <LivingRoomLight>1</LivingRoomLight>
    <FrontDoorLight>0</FrontDoorLight>
    <KitchenLight>0</KitchenLight>
    <TempratureReading>30</TempreatureReading>
 </response>
```

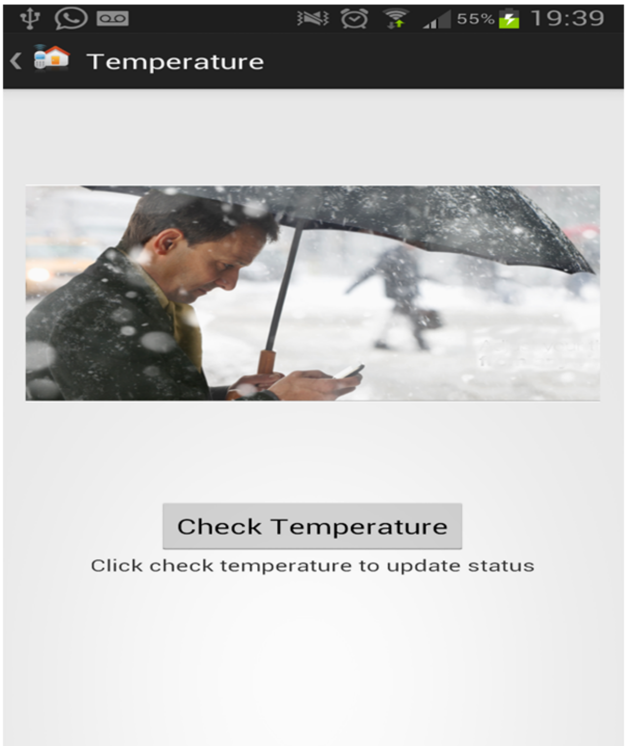
Whenever the android app received an XML file contacting updates xmlParse function is called. The purpose of the function is update user interface to match the latest status update received.

```java
private void parseXML(String myXML) throws XmlPullParserException, IOException
{
    String ss="";
    XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
    factory.setNamespaceAware(true);
    XmlPullParser xpp = factory.newPullParser();

    xpp.setInput(new StringReader (myXML));
    int eventType = xpp.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT) {
        if(eventType == XmlPullParser.START_TAG) {
            ss+=xpp.getName();
            if (xpp.getName().compareTo("LivingRoomLight")==0){
                eventType = xpp.next();
                if(eventType == XmlPullParser.TEXT) {
                    boolean status1 = false;
                    if (xpp.getText().compareTo("1") == 0)
                        status1 = true;
                    localCall= false;
                    s1.setChecked(status1);

                    localCall = true;

                }
```

The function would loop the file from beginning to the end. In each loop it reads a new tag (Start_Tag). It compares that tag text with LivingRoomLight as illustrated in the code snapshot. If it matches it will move next to read the inner text which represents the status of the living room light. Then the appropriate switch on the user screen would be set approrately using setChecked method. It is worth mentioning here that we need to set the global variable localCall to false before changing the status of the switch to avoid a loop between the toggle and status feedback. That is because the toggle will call the switch listener and request from the PIC32 a change in the light status.

The second activity (user screen) is used to display temperature.

Whenever the user click on check temperature, again using the same method discussed above an http request is sent to PIC 32 and the feedback is XML status containing the temperature update.



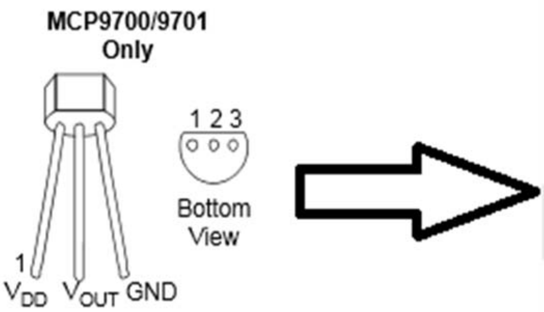
The third user screen is the configuration screen. It is used to set the IP address of the PIC 32 controller on the network. This is needed to insure communication between the android app and PIC32 over IP protocol. The IP address is stored in global preferences so that other activities in android, namely the lights and temperature activities would have access to the IP address.

Sensors:

Temperature sensor- MCP9700- Microchip's Temperature Sensor

We have used an analogue temperature sensor which is connected to PIC microcontroller which can sense the temperature of house. By using PIC microcontroller we can transmit that temperature reading to mobile phone using Wi-Fi. Temperature sensor is connected to 2, 3, 7 analogue pin of PIC microcontroller.

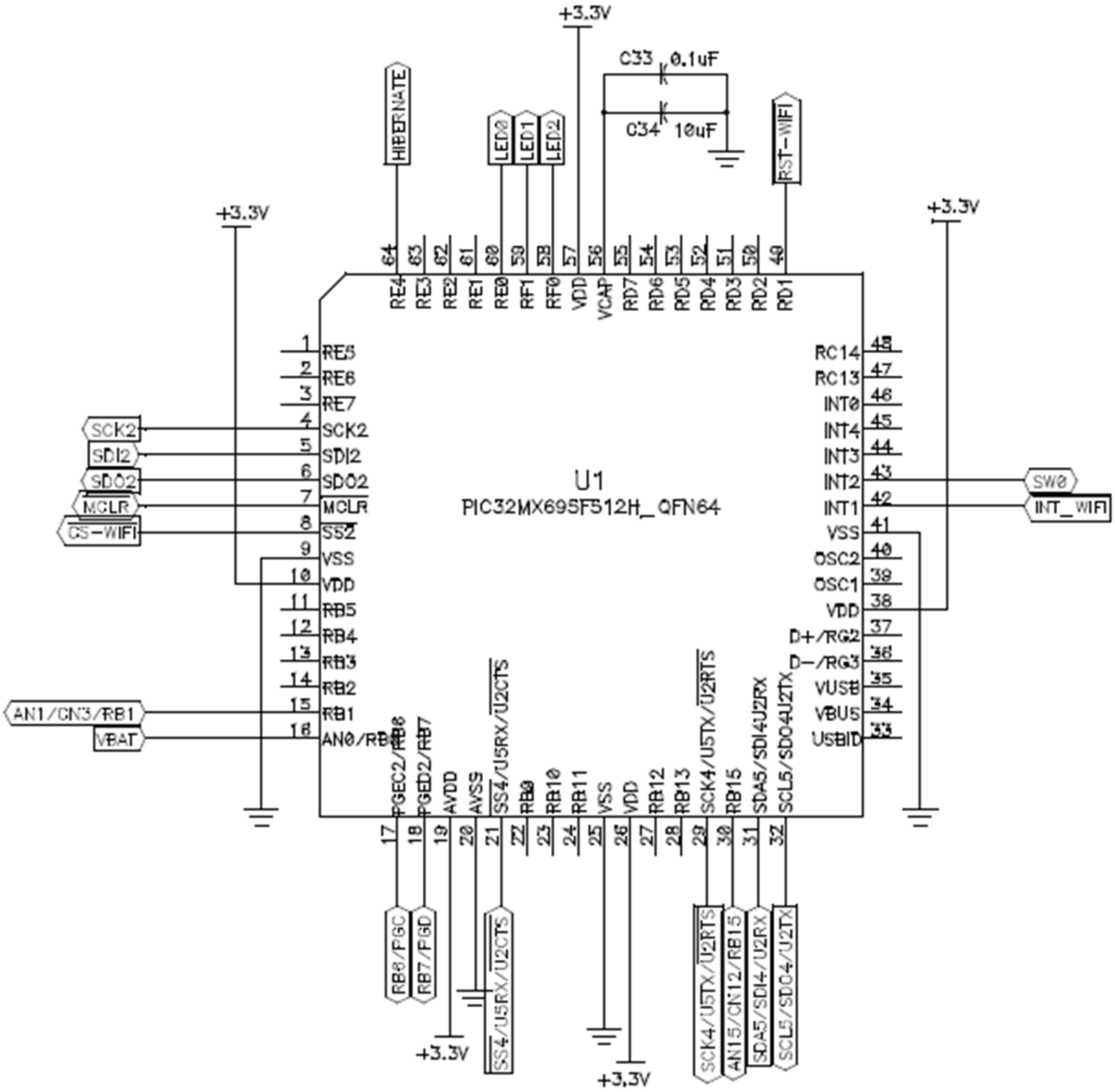
Pictorial representation of temperature sensor and its features:

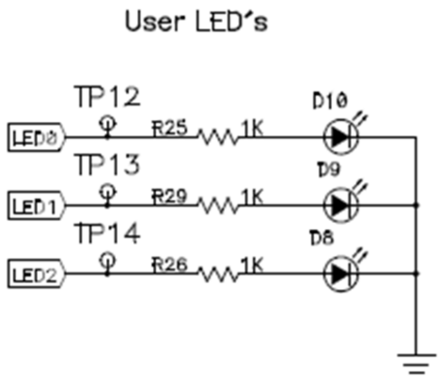


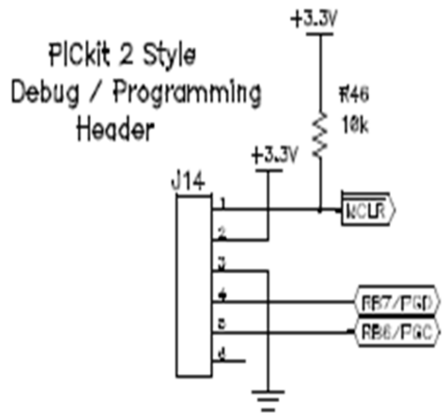
Features

1. Analog Temperature Sensor
2. Wide Temperature
3. Measurement Range:
    a. -40°C to +125°C (Extended Temperature)
    b. -40°C to +150°C (High Temperature)
4. Accuracy:
    i. ±2°C (max.), 0°C to +70°C
5. Optimized for Analog-to-Digital Converters (ADCs):
    i. 10.0 mV/°C (typical)
6. Wide Operating Voltage Range:
    a. VDD = 2.3V to 5.5V
7. Low Operating Current: 6 µA (typical)
8. Optimized to Drive Large Capacitive Loads

Schematic diagram of PIC microcontroller:

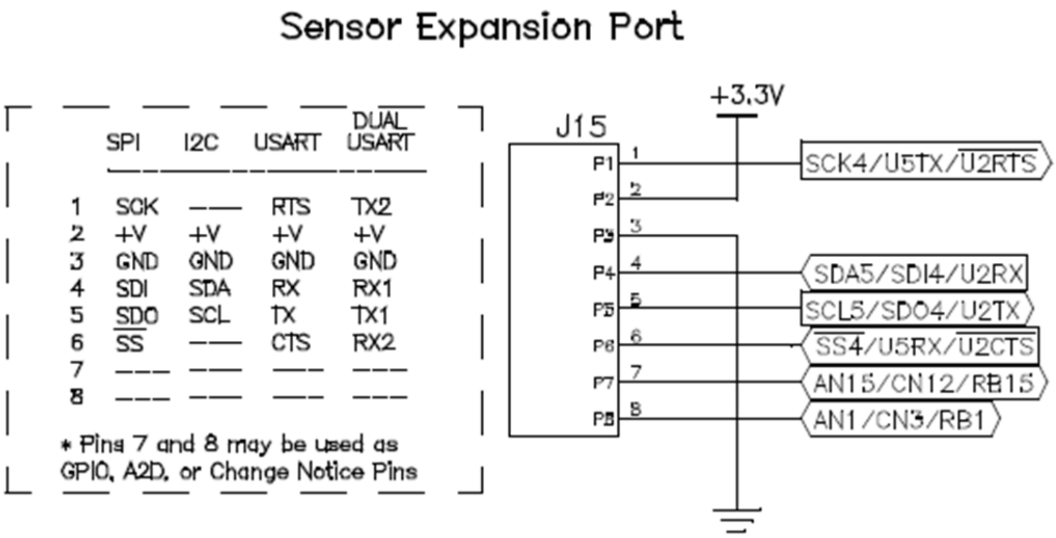




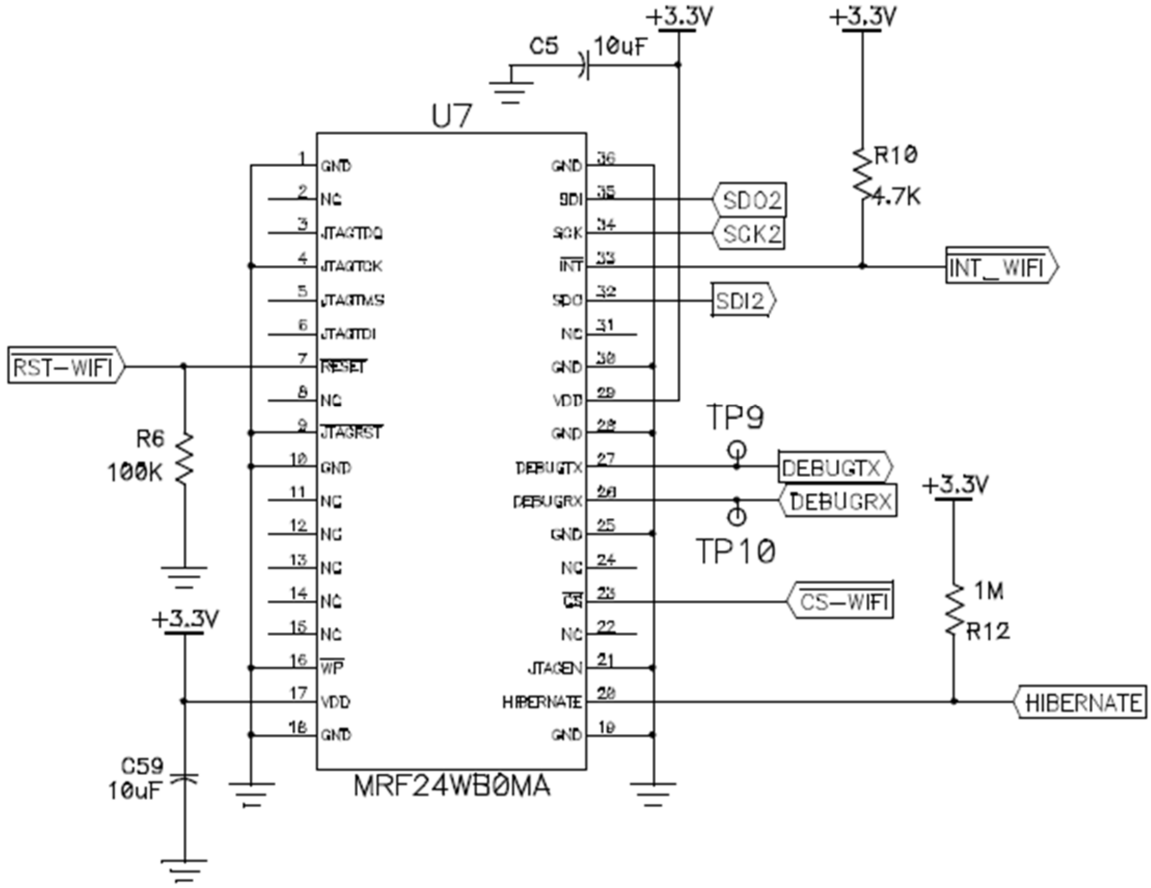


LED connectivity to board

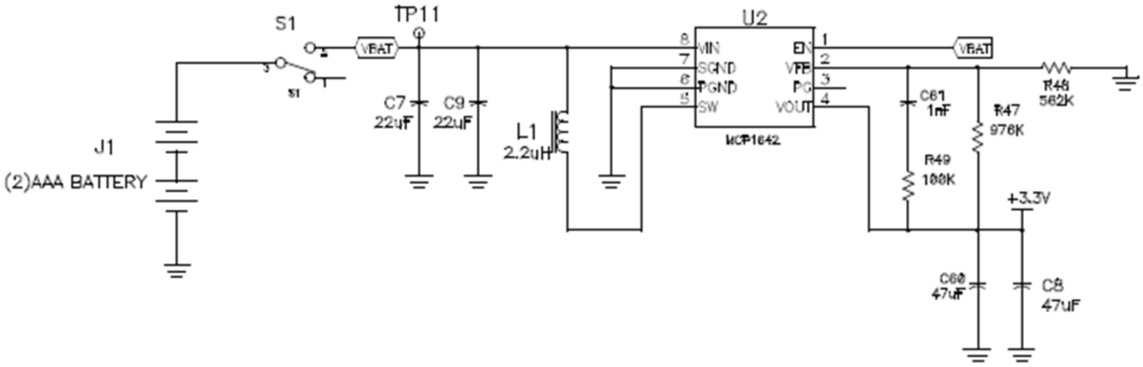
Sensor expansion board:



On-Board Wi-Fi:



Power Supply:



Analysis:

For controlling lights in house you can program PIC microcontroller that is connected to house and you can control that PIC by your phone using A Wi-Fi connection, using smart home system you can control lighting and temperature throughout your home remotely from a smartphone. You can dim lights while watching a movie or having dinner, or schedule lights to turn on and off while you are on vacation. Stay green and save energy by checking to see if any lights were left on, or turn off anything that is plugged in. Consider using a programmable thermostat (future expansion) to increase energy management and reduce your utility bill by regulating thermostat use and temperatures based on your family's schedule.

Result:

Our goal was to implement smart home system in which we can turn ON and OFF lights using our smart phone and also we can sense temperature of home on our smart phone. This objective is achieved successfully though we faced some issues but those issues were resolved (discussed later in this report).We can successfully turn lights ON and OFF and sense temperature through phone. It is easy to expand this project to install additional features to include voice control, opening doors through phone, media control, and much more in security like installing camera for surveillance.

Conclusion:

The project has been completed successfully. All sub-objective of this project has been completed, thus resolved the main objectives and problems that occur during this project. This system managed to help users to detect where the light is required and sensing temperature inside house even if you are not there thus reduce their effort. The use of wireless connectivity based on microcontroller facilitates the installation process at home and makes it more flexible and easy for user. This system is a system that meets recently smart home applications in order to function in automation situation. During the study, a lot of knowledge and experience have been learned. Besides the exposure on microcontroller, this study also has much exposure to android programming through which we can create an application for smart phone for receiving and sending signal to PIC microcontroller, where it helps in the transfer process for the wireless signal.

References:

1. Microchip Website http://www.microchip.com/
2. http://www.microchip.com/forums/m565422-print.aspx
3. Xfinity Comcast Website:http://www.comcast.com/home-security?CMP=KNC-IQ_ID_62226835-VQ2-g-VQ3--VQ6-45533632065-VQ16-c-pkw-%2Bsmart%20%2Bhome-pmt-b&iq_id=62226835
4. Smart Home: https://courses.cit.cornell.edu/ee476/FinalProjects/s2010/jps89_cdq2_mam584/jps89_cdq2_mam584/index.html
5. http://ece.eng.wayne.edu/~smahmud/ECECourses/ECE5620/ECE5620.htm

Appendix:

Executive Summary:

A Smart Home is one that provides its home owners comfort, security, energy efficiency (low operating costs) and convenience at all times, regardless of whether anyone is home. As the cost of living is increasing, there is a growing focus to involve technology and develop smart systems.

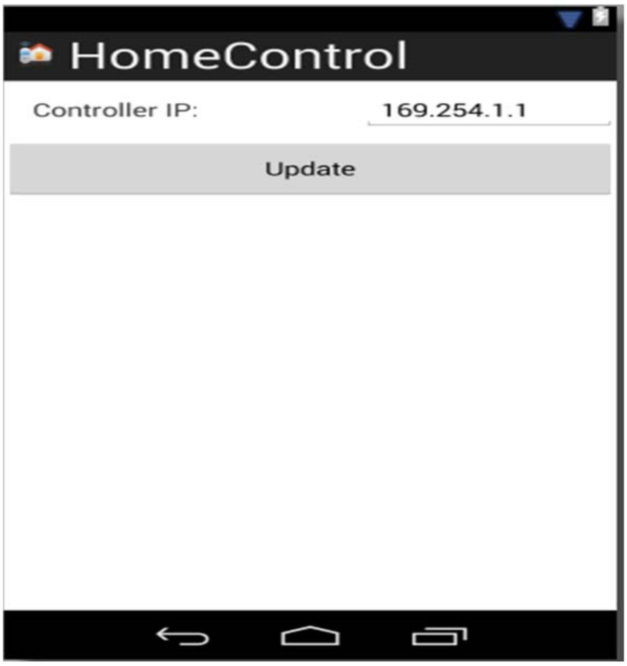
An emerging important feature of a smart home is conservation of the earth's limited resources. More and more people are becoming aware of the ability to make their homes truly smart and eco-friendly by utilizing home controllers integrated with all home sub-systems to increase savings by controlling lighting, window coverings, HVAC, irrigation and by monitoring usage. Many home controllers have built-in monitoring systems whereby they calculate and log usage by all connected devices, giving the home owner heightened awareness and the knowledge to make changes as necessary. These systems can even be accessed over the Internet from anywhere in the world so the homeowner can adjust consumption anytime, anywhere. With a home that has this advantage, you can know that your home is performing at its best in energy performance.
By implementing this project we were able to explore a variety of different engineering challenges, including software programing, PCB design, Wi-Fi, TCP/IP protocols, Web Server logic design, and other aspects. This project provided great insights to the challenges of software and hardware engineering.

The 32bits-PIC Microcontroller is the core of the system, and the Web Server with Combining with CGI (Common Gateway Interface) Technology, it is implemented to exchange data and information between the Server and remote users. And Wi-Fi wireless module is to used exchange data and information between information appliances and the Server.

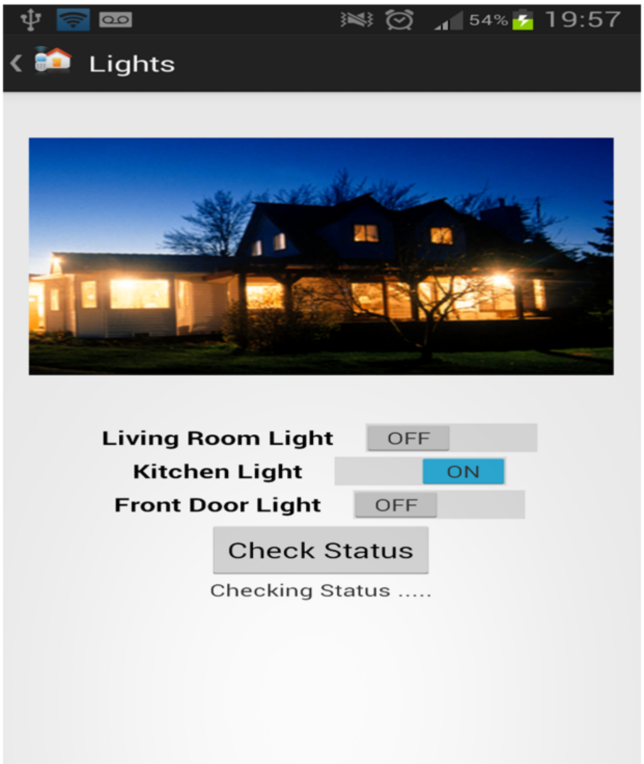
Design Alternatives:

1. Implementing the same using PIC18 Family processor and Using Wi-Fi mode for Control.
2. Use of Atmel processor, wireless mode of control and use of Aurdino programming

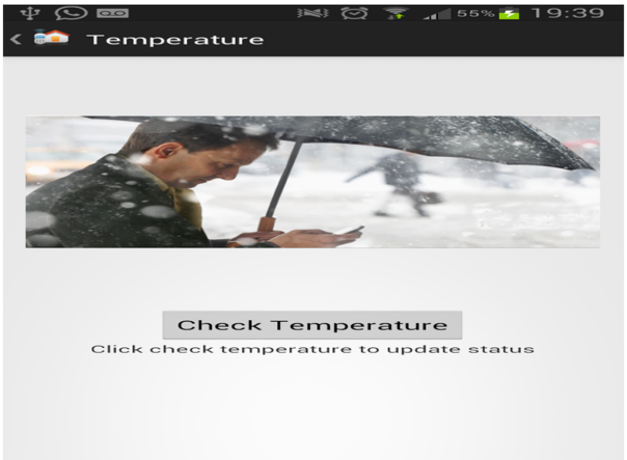
Operating Procedure:



Setup the IP address as configured in pic32 code in android app



Navigate to Lights control and click any switch or request to check status



Navigate to temperature control and click Check Temperature

Federal Communications Commission (FCC) Rules:

We have used the Microchip's RF Transceiver chip to set up the wireless communication. To make sure if we could use the RF transceiver to set up the wireless communication in the United States, we need to check the Federal Communications Commission Rules. This Transceiver is certified by Radio regulation certification for United States (FCC), Canada (IC), Europe (ETSI) and Japan (ARIB). Transceiver is operated in the ISM Band of 2.400-2.483.5 GHz operation frequency.
After we check the FCC rules part 15 subpart B, we got 2.4 GHz frequency are exempt from complying with the technical provisions of this part which prevent the harmful interference in a residential installation. For more details refer to the "70632A RF transceiver Datasheet" Manual.

Distribution of work among different partners of a group, and the contribution of each partner in percent:

Android App: Iyad Kuwatly 100%
Pic32 Code and hardware interfacing: Pavankumar Bhujannavar 40%, Shivesh Kumar Sharma 40%, Iyad Kuwatly 20%
Integration and testing: Iyad Kuwatly 33% , Pavankumar Bhujannavar 31%, Shivesh Kumar Sharma 36%
Report editing: Iyad Kuwatly 13%, Pavankumar Bhujannavar 40% and Shivesh Kumar Sharma 47%

Problems occurred during the design process and how the problems were solved:

Some of the problems in the design process include the following:

1- Unable to find the pic32 Wi-Fi network from android phone

Using lap top, we were able to scan all available nearby Wi-Fi networks and after research on the internet, we figured out that Android does not support Wi-Fi Adhoc mode and that is why the pic was not directly visible and thus we had to add a Wi-Fi and work on Wi-Fi infrastructure mode.

2- A problem with data being sent from pic 32 was received as garbage.

The reason is that we needed to send data in http using ASCII characters and that also applies to numbers as well. For example integers 0 need to be sent as decimal 49 and html &#49. The following table illustrates the mapping:

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040		Space	64	40	100	@	@	96	60	140	`	`
33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Parts List and Cost Analysis:

Part/ Module	Manufacturer	Quantity	Price
53C35K Potentiometer	Honeywell	1	$7.79
C513A-WSN-CV0Y0151 LED	Cree Inc.	10	$2.31
MCP9700A-E/TO Temperature Sensor	Microchip	2	$0.62
DV102411 Wi-Fi demo board	Microchip	1	$49.99
Connecting Wires		1	$0.50
Crocodile clips		6	$0.25
Electric Tape			
Shipping charges		2	$15.00
		Total	$ 76.46

Datasheets:

70632A RF transceiver Datasheet
C513A WSN WSS MSN MSS 1042 LED DATASHEET
MTTempIC- Datasheet
PIC32MX- Datasheet
Wi-Fi Comm Demo User Guide_70678A- Datasheet

Parts/ Modules purchased from: Newark element14

Program Listing

The software code of the project is found in 2 sections:

1. PIC programming
2. Android Programming

Acknowledgement:

Our first and sincere appreciation goes to Professor Syed Mahmud for all we have learned from him and for his continuous help and support in all stages of this project. We would also like to thank him for being an open person to ideas, and for encouraging and helping us to shape our interest and ideas.

In addition, we would like to thank TA Mr. Yunrui Li for his help in the lab and answering our questions promptly.