“WIRELESS SENSOR ADHOK NETWORK”

(Phase 2 Report)



**Submitted By**

**WAJID ALI KHAN 24600**

**SYED NISAR SHERAZI 24592**

**MUHAMMAD TARIQ KHAN 24569**

**SYED NADEEM SHERAZI 24591**

**Supervisor**

**SIR FAHEEM AWAN**

BS Telecommunication Department of Information Technology

Hazara University Mansehra

**April 13, 2015**

**Introduction:**

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous tiny computing devices, each equipped with a wireless radio, a processor, and a power source. In this project we are adding an intelligence factor. Intelligence factor means that the sensors will be connected to the main circuit where all the calculations will be carried out and then the sensors data will be forwarded wirelessly to the intermediate node and then the intermediate node will decide where to send the each sensor data. Sensor networks are envisioned to be deployed in the physical environment in order to monitor a wide range of environmental phenomena.

Previously, sensor networks consisted of small number of sensor nodes that were wired

to a central processing station. However, nowadays, the focus is more on wireless, distributed, sensing nodes. In most cases, the environment to be monitored does not have an

existing infrastructure for either energy or communication. It becomes imperative for sensor nodes to survive on small, finite sources of energy and communicate through a wireless

communication channel.

Sensor network application like weather monitoring needs data like temperature, light intensity, and motion detector etc. called as sensing modalities, there exist different sensors to sense each sensing modality.

Work on project:

Before starting the work on project we have divided our project in four different phases, firstly we will complete all phases on software i.e. Proteus and CADSOFT EAGEL (Proteus for simulation and CADSOFT EAGEL SOFTWARE for PCB designing). In last phase we will design the project on hardware.

The phases of this project are as below.

Phase (1)

Phase (2)

Phase (3)

Phase (4).

Phase (1) has been completed. Now we are going to show the report on phase (2).

**“PHASE 1”**

**Block diagram:**

12 v

zigbee

controller

Light Intensity

Temperture

In this phase a 12v DC power supply is connected to the At mega controller with this power controller will start working. Now at the lower side of this figure we can see that two sensors are connected to the At mega controller, first one is temperature sensor which will sense the temperature from the room and send it to the At mega controller. Secondly light intensity sensor will sense light of the room and send it to At mega controller. As At mega controller collects the data from temperature and light intensity sensor it will transfer its data to intermediate node through zigbee.

“PHASE 2”

Block Diagram:

zigbee

Intermediate Node

controller

12 v

Load 2

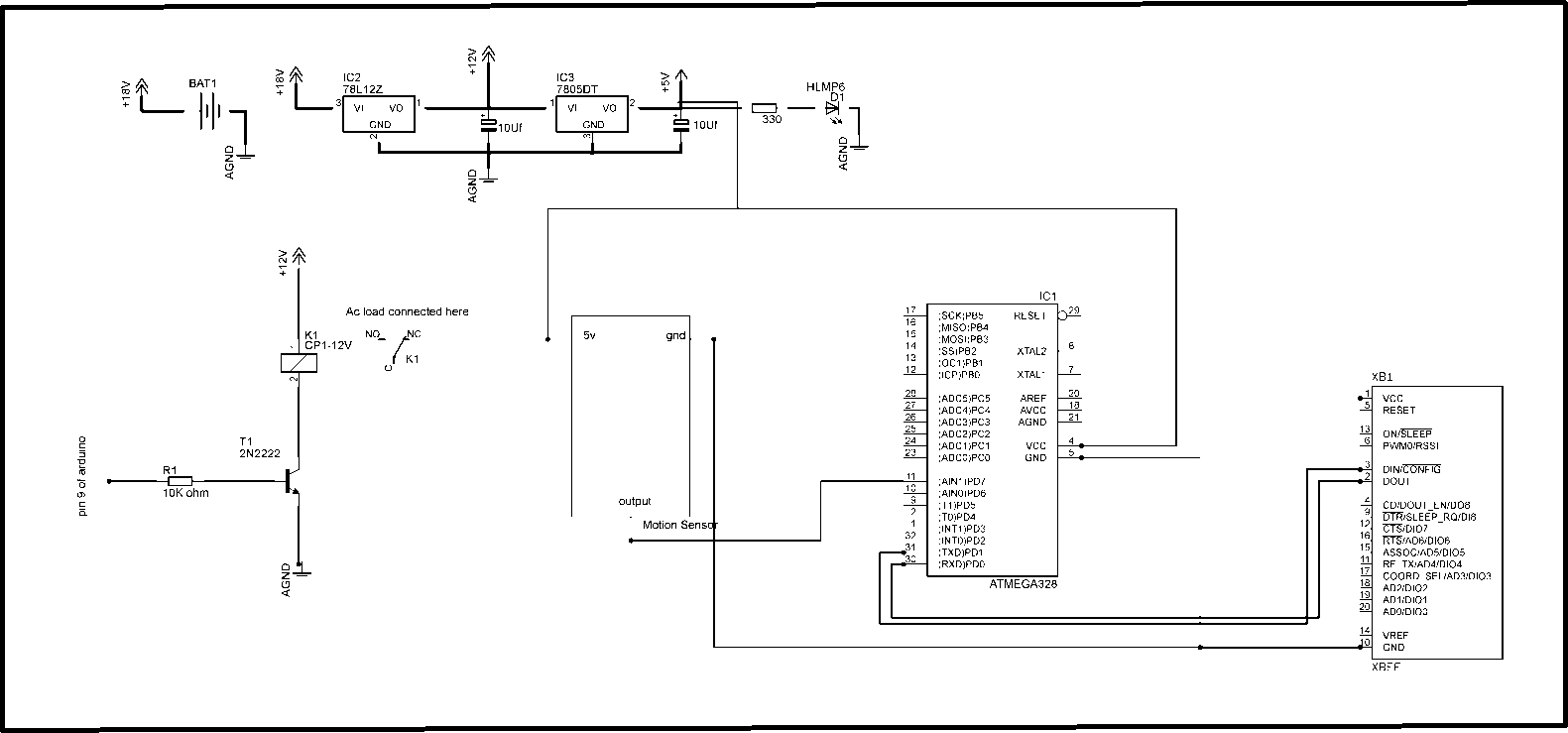
Motion sensor

Load 1

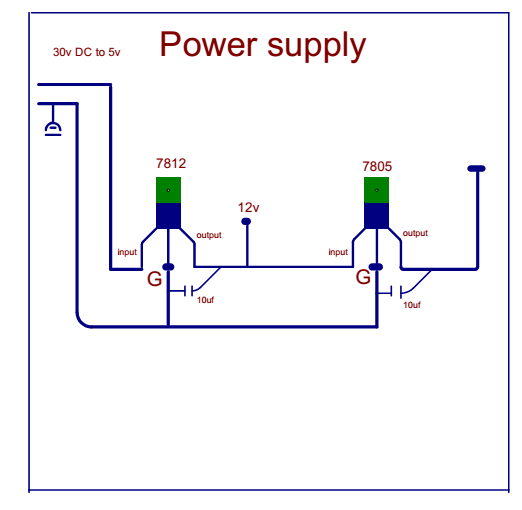
In phase 2 zigbee will receive the data from phase one and give it to the intermediate node (controller). A 12v DC power supply is connected to controller to give the power to the controller. We have connected the load 1 and load 2 to intermediate node now when the intermediate node receives the data from zigbee it will transfer it to load 1 and load 2. We have also connected a motion detector sensor to intermediate node, we have supplied 5v power to motion sensor so that it can sense the motion in the room and sent the data to controller and then controller will send this data to another controller through zigbee .

**Work on phase 2:**

This is complete diagram of phase 2 which is been made by CADSOFT EAGEL SOFTWARE.



**Power supply:**



In this power supply we use 2 voltage regulator 7812 and 7805. 7812 which converts 30 DC volt to 12 volt and 7805 which will convert 12 volt into 5 volt. Here 5 volt is taken for motion sensor that cannot operate more then 5 volt.

**Atmega Controller:**



The Atmega [8 bit](http://en.wikipedia.org/wiki/8-bit) [AVR](http://en.wikipedia.org/wiki/Atmel_AVR) [RISC](http://en.wikipedia.org/wiki/RISC)-based microcontroller combines 32 KB [ISP](http://en.wikipedia.org/wiki/In-system_programming) [flash](http://en.wikipedia.org/wiki/Flash_memory) memory with read-while-write capabilities, 1 KB [EEPROM](http://en.wikipedia.org/wiki/EEPROM), 2 KB [SRAM](http://en.wikipedia.org/wiki/Static_random-access_memory), 23 general purpose I/O lines, 32 general purpose working [registers](http://en.wikipedia.org/wiki/Processor_register), three flexible timer/[counters](http://en.wikipedia.org/wiki/Counters) with compare modes, internal and external [interrupts](http://en.wikipedia.org/wiki/Interrupts), serial programmable [USART](http://en.wikipedia.org/wiki/USART), a byte-oriented 2-wire serial interface, [SPI](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus) serial port, 6-channel 10-bit [A/D converter](http://en.wikipedia.org/wiki/A/D_converter) (8-channels in [TQFP](http://en.wikipedia.org/wiki/TQFP) and [QFN](http://en.wikipedia.org/wiki/QFN)/[MLF](http://en.wikipedia.org/wiki/Quad-flat_no-leads_package#Variants) packages), programmable [watchdog timer](http://en.wikipedia.org/wiki/Watchdog_timer) with internal [oscillator](http://en.wikipedia.org/wiki/Electronic_oscillator), and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughputs approaching 1 [MIPS](http://en.wikipedia.org/wiki/Million_instructions_per_second#Million_instructions_per_second) per MHz.

**Simulation:**

As we have connected motion sensor in phase 2 to detect motion. as we are using Proteus software for simulation, in Proteus software doesn’t have module of motion sensor that is why simulation of phase 2 is not possible. We are searching for such software in which motion sensor module is present so that simulation of this phase becomes possible.

**Arduino Programing:**

#include <VirtualWire.h>

#include <SoftwareSerial.h>

SoftwareSerial Xbee(2, 3);

const int led = 13;

const int transmit\_pin = 12;

int motion = 7;

const int transmit\_en\_pin = 3;

int bulb = 9; // bulb connected here

char data ;

boolean flag = 0;

boolean flag2 = 0;

String str;

char cstr[27];

void setup()

{

// Initialise the IO and ISR

vw\_set\_tx\_pin(transmit\_pin);

Serial.begin(9600);

Xbee.begin(9600);

vw\_setup(2000); // Bits per sec

pinMode(led, OUTPUT);

pinMode(motion, INPUT);

digitalWrite(led, LOW);

pinMode(bulb, OUTPUT);

}

byte count = 1;

void loop()

{

if(Xbee.available()>0)

{

data = Xbee.read();

} else

if((digitalRead(motion) == 1)&& (flag2 == 1))

{

digitalWrite(bulb , HIGH );

}else

if(data == 'a')

{

digitalWrite(bulb, HIGH);

flag = 1;

}else

if(data == 'b')

{

digitalWrite(bulb, LOW);

flag = 0;

} else

if(data == 'o')

{

digitalWrite(bulb, LOW);

flag2 = 0;

}

else

if(data == 'n')

{

//digitalWrite(bulb, HIGH);

flag2 = 1;

}

else

if((digitalRead(motion) == HIGH) && ( flag == 1) && (flag2 == 1))

{

digitalWrite(bulb, HIGH);

flag = 0;

flag2 = 0;

} else

if((digitalRead(motion) == LOW) && (flag == 0)&& (flag2 == 0))

{

digitalWrite(bulb, LOW);

flag = 1;///////

flag2 = 1;//////

}else

if((digitalRead(motion) == HIGH)&& (flag == 0)&& (flag2 == 0))

{

digitalWrite(bulb, HIGH );

flag = 1;//////

flag2 = 1;/////

//delay(4000);

//digitalWrite(bulb, LOW);

}else

if((digitalRead(motion) == HIGH) && (flag2 == 1))

{

digitalWrite(bulb , HIGH);

flag2 = 0; ////////////////////////////////////////

}

//else

//digitalWrite(bulb, LOW);

}