Analog Communication Project Report

Design an Arduino Application for remotely sensing and communicating the sensed signal using Arduino and XBee Module.

GROUP-10

Group Members:
K.Mohith Prakash-22EC01002
Y.Harsha Vardhan-22EC01007
Vikram Patil-22EC01008
G.Hari Krishna-22EC01011

ABSTRACT

Remote sensing is one of a suite of geospatial technologies that are having a growing impact in a wide variety of areas from commerce to science to public policy. The field of remote sensing evolved from the interpretation of aerial photographs to the analysis of satellite imagery, and from local area studies to global analyses, with advances in sensor system technologies and digital computing. Today remote sensor systems can provide data from energy emitted, reflected, and/or transmitted from all parts of the electromagnetic spectrum. Examples of applications of these data include population and demography studies, study of archaeological sites, energy studies using hydrological models, urban planning, environmental treaty enforcement, and agricultural yields.

INTRODUCTION

In today's era of interconnected devices, the demand for remote monitoring and data communication systems has surged across various fields. Whether it's environmental monitoring, industrial automation, or smart infrastructure, the ability to remotely sense and transmit data is indispensable for efficient operations and decision-making.

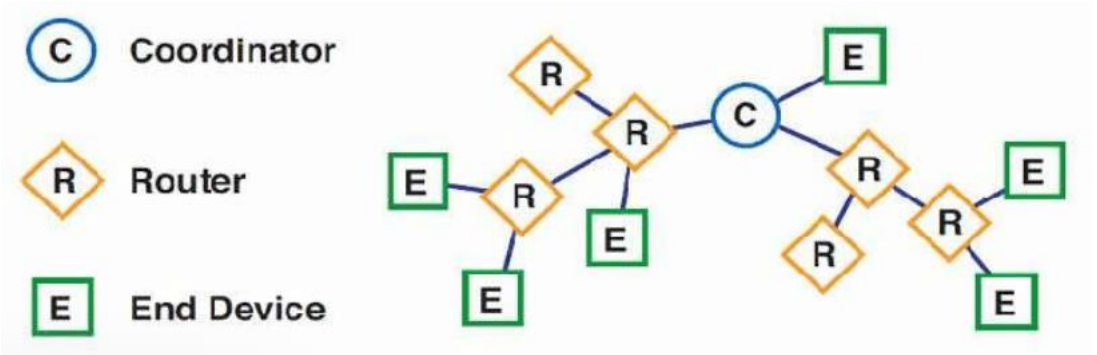
Our project focuses on developing a robust and versatile solution for remote sensing and communication using Arduino microcontrollers and XBee modules. Specifically, we utilize an IR (Infrared) sensor for data acquisition and XBee modules for wireless communication over a ZigBee network.

OBJECTIVE

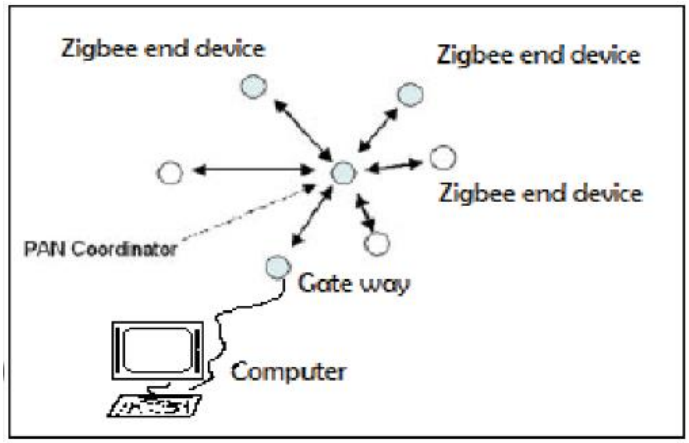
The primary objective of this project is to design a system capable of:

- Remote sensing: Implementing an IR sensor to detect changes in infrared radiation, typically used for proximity sensing or object detection.
- Wireless communication: Establishing communication between multiple Arduino nodes using XBee modules, facilitating the transmission of sensor data over a ZigBee network.
- Real-time monitoring: Enabling users to remotely monitor sensor readings from any location with access to the ZigBee network

NETWORK DESIGN



For this experimental study, we propose a network based on the Arduino development platform with a mesh topology. A setup of the proposed network is depicted in the figure below. We will consider this network as static. The end nodes will be tasked with sensing infrared (IR) values and transmitting this information to the coordinator radio. The coordinator node serves as a pivotal and powerful node within the network, akin to a sink node, responsible for gathering all transmitted data. Routers, positioned as intermediate devices, are tasked with relaying or forwarding packets within the network. Sensor nodes function as endpoint devices, periodically entering sleep mode to conserve energy.



HARDWARE SPECIFICATIONS

1. ARDUINO UNO

The Arduino Uno simplicity and versatility make it ideal for both beginners and advanced users. With its easy-to-use interface and a wide range of compatible sensors and modules, the Arduino Uno can collect data from its surroundings, such as temperature, humidity, or light intensity. When paired with an XBee module, it can wirelessly transmit this collected data over long distances, enabling remote monitoring and control.

2. XBEE MODULE

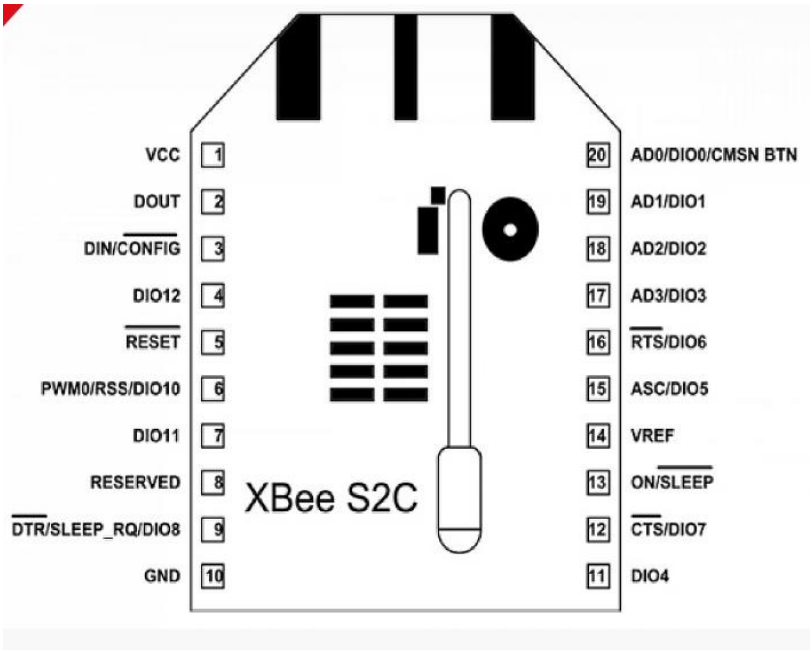
The XBee module is like a tiny radio that helps Arduino communicate wirelessly over long distances. It's a small but powerful device that allows you to send and receive data without needing any physical wires. Imagine it as a messenger bird that carries your messages back and forth between your Arduino and another Arduino or computer. With the XBee module, you can build projects like remote sensing, where you can collect data from sensors placed far away and send that data to your Arduino for processing. It's a handy tool for making projects that need to communicate without being physically connected.

In remote sensing applications, each node communicates wirelessly with one another. XBee modules serve as the communication backbone for such setups. Despite common misconceptions, ZigBee and XBee are not synonymous terms. XBee, in fact, constitutes ZigBee compliant hardware [3]. ZigBee, on the other hand, represents a standard communications protocol tailored for low-power, low-throughput, and low-cost wireless mesh networking applications.

1. XBee Series 1 hardware: These radios integrate a microchip manufactured by Free scale, offering straightforward, standards-based point-to-point communications alongside a proprietary implementation of mesh networking.

2. XBee Series 2 hardware: This series adopts a microchip from Ember Networks, enabling various flavors of standards-based ZigBee mesh networking. Both Series 1 and Series 2 radios are available in two different transmission powers. The regular transmission power version is simply referred to as an XBee.

In this setup, we've utilized an XBee S2C Module operating within the 2.4 GHz frequency band.

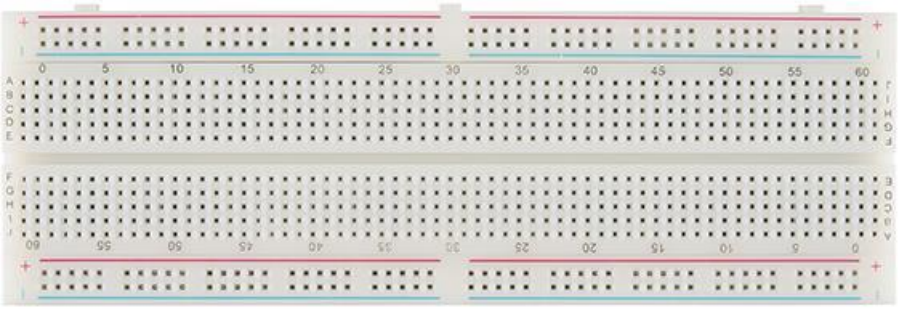


3. **INFRARED SENSOR**

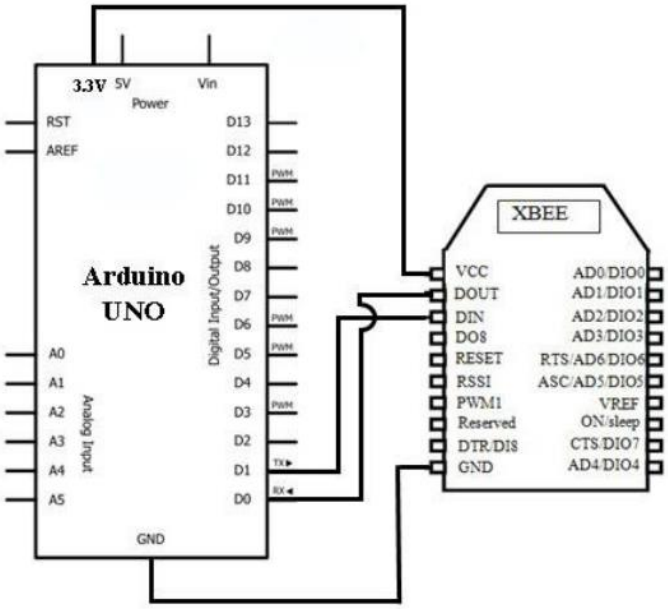
IR sensors, short for Infrared sensors, are devices that can detect infrared radiation, which is invisible to the human eye. In the context of the experiment IR sensors play a crucial role in detecting objects or obstacles without direct contact. These sensors emit infrared light and measure the reflection or presence of this light to determine the proximity or absence of an object. When integrated with Arduino and Xbee Modules, IR sensors enable the creation of a remote sensing system that can detect and communicate information wirelessly, making it ideal for applications such as security systems, automated robotics, or smart home devices.

4.BREADBOARD

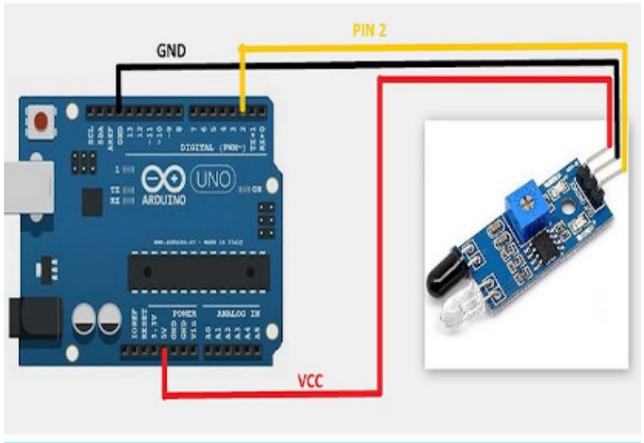
A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. They consist of a plastic housing riddled with small holes. Metal clips that lurk beneath the holes in the breadboard provide a way to hold and connect components.



CIRCUIT DIAGRAM



ARDUINO AND XBEE CIRCUIT



ARDUINO AND IR SENSOR CIRCUIT

BASIC WORKING PROCESS

The experimentation process unfolds in a series of sequential steps:

1.Initial Setup of Arduino and XBee: Commences with configuring the Arduino board and XBee module, establishing their interconnectivity, and ensuring seamless communication between them.

2.Sensor Integration: Proceeds with linking the sensor—detecting parameters like temperature or light to the Arduino board. This sensor serves as the data source for remote monitoring.

3.XBee Configuration:Involves configuring the XBee modules to enable wireless communication between the Arduino and another remote device. This encompasses

tasks such as setting transmission frequency and network identification.

4.Arduino Code Development:In this phase, programming code is written for the Arduino to gather data from the sensor. The code instructs the Arduino on how to process and transmit the collected sensor data.

5.Data Transmission: The Arduino collects data from the sensor and transmits it to the XBee module. Subsequently, the XBee module wirelessly transmits this data to its designated receiving end.

6.Data Reception:At the receiving end, another XBee module intercepts the wirelessly transmitted data. This receiving module may be installed on another Arduino or a computer, equipped with an XBee module.

7.Data Visualization and Analysis: Concludes with the display or analysis of received data on the receiving device. This step facilitates real-time monitoring or in-depth analysis of the sensor readings remotely.

In essence, the experimental process encompasses the integration of an Arduino with a sensor, utilizing XBee modules for wireless transmission of sensor data to a remote device, and subsequently visualizing or analyzing the received data at the receiving end.

CONFIGURATION OF XBEE FOR BASIC COMMUNICATION

To configure the XBee Series 2 radio for basic communication, follow these steps:

1. Connect the XBee Series 2 radio to the XBee Explorer board, and then plug the Explorer board into one of the USB ports on your computer.

2. Launch the X-CTU software. Switch to the "Modem Configuration" tab, which will populate a window displaying various useful information.

3. The XBee should be automatically detected, and its model type should be listed as XB24-ZB.

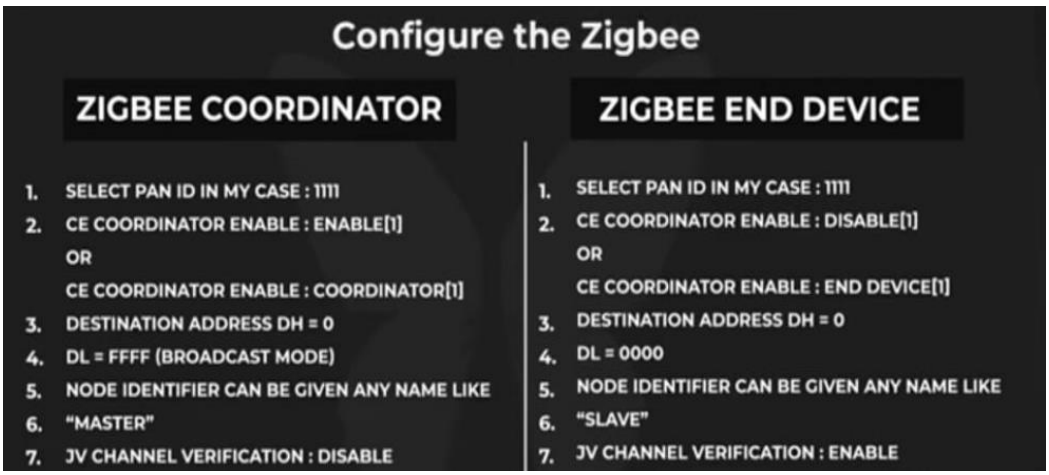
4. Under the "Function Set" section, you'll see a list of different firmware options available for this class of radio. Depending on the intended function of the modem in your network, you can select options such as ZigBee Coordinator AT or ZigBee Router AT.

5. Other important parameters that may need to be configured include PANID (Personal Area Network ID), Destination Address High, and Destination Address Low.

6. Click the "Write" button to program the radio with the selected configurations.

7. You can also use AT commands to check and configure radio addresses and other parameters as needed.

CONFIGURATION SETTINGS OF XBEE IN XCTU



 Naming of both Xbee is done.
Xbee is set to either coordinator mode or End device mode.

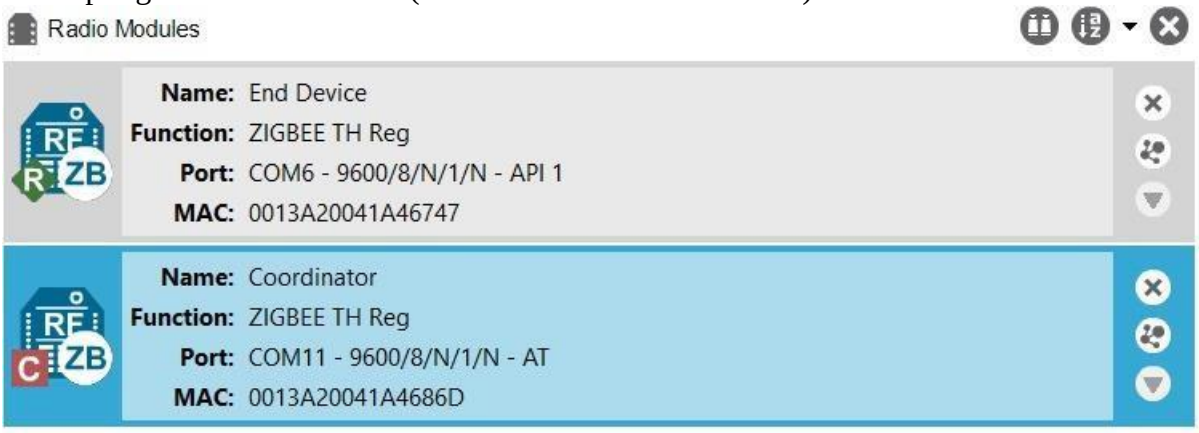
 API mode enabled for coordinator and disabled (AT mode) for End device.

- Transparent mode (AT command)-In Transparent mode, whatever data is available on DIN pin is directly transmitted to receiver (in case of point to point) or receivers (in case of point to multipoint).
- Application Programming Interface (API) mode -In API mode, data is wrapped in frame. Frame consists of Start Delimiter, Frame Length, Frame Type, Data, Checksum etc. Parameter setting and packet delivery feedback can be viewed in API mode.

 PAN ID and address of both Xbee are kept same to establish a connection between them.

 Pin D3 of End device Xbee is set to ADC(Analog to Digital Converter) to retrieve analog values from IR SENSOR.

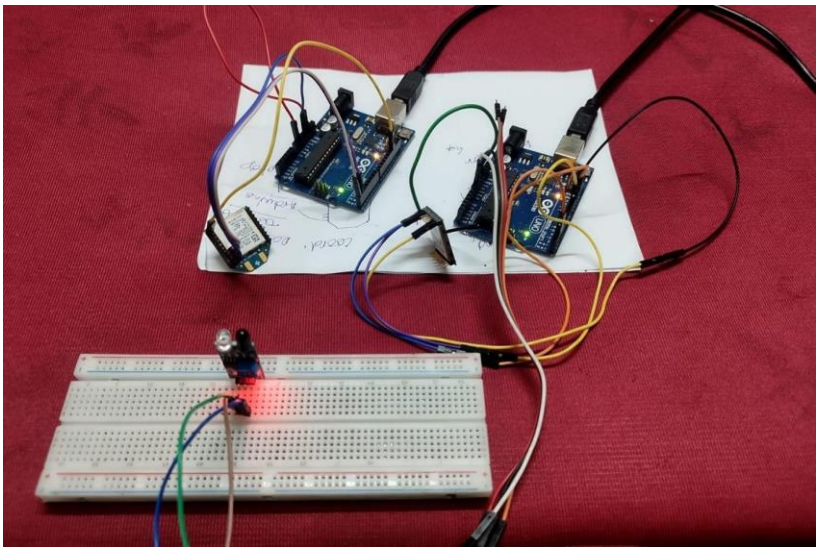
 Sampling rate is in hex value (1388 in hex =5000 in decimal)



ARDUINO IDE CODE

```
#include <IRremote.h>
#include <SoftwareSerial.h>

// IR sensor setup
const int IRPin = 2; //  IR sensor connected to digital pin 2
IRrecv irrecv(IRPin);
decode_results results;

//  XBee setup using software serial
SoftwareSerial XBee(10, 11); // RX, TX

void setup()
{
  Serial.begin(9600);
  XBee.begin(9600);
  irrecv.enableIRIn(); // Start the IR receiver
}

void loop()
{
  // Check if the IR sensor has received data
  if (irrecv.decode(&results))
  {
    // Send the received IR data over XBee
    XBee.println(results.value, HEX);
    Serial.println(results.value, HEX);
        irrecv.resume(); // Receive the next IR signal.
  }
}
```

WORKING MODEL



CONCLUSION

Through this project, we aim to showcase the practical implementation of remote sensing and communication systems using widely available hardware components and open-source software tools. By leveraging Arduino and XBee technologies, our system offers a flexible and scalable solution for remote monitoring applications, with potential applications in various domains, including smart agriculture, industrial automation, and infrastructure management.

This experiment underscored the importance of hands-on learning in understanding the fundamentals of electronics and wireless communication, equipping us with valuable skills for future endeavors in the field.