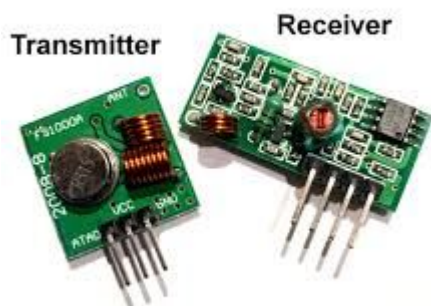


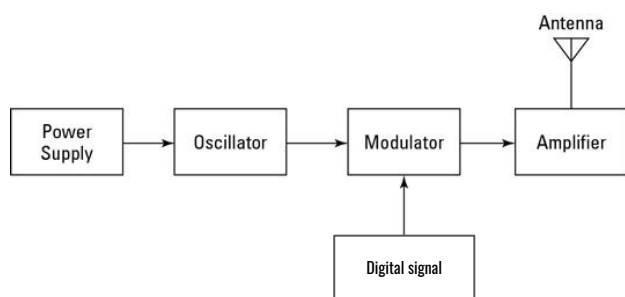
Introduction



In the 19th century, scientists discovered how to electronically generate radio waves using electric currents for communication. Two components are required for radio communication: a transmitter and a receiver. A **radio transmitter** generates radio waves that contain useful information such as audio, video, or digital data. A **radio receiver** uses an antenna to capture radio waves, processes those waves to extract only those waves that are vibrating at the desired frequency, extracts the analog or digital signals that were added to those waves. An AM/FM radio is an example of an analog receiver. A modern mobile phone is an example of a digital receiver and transmitter.

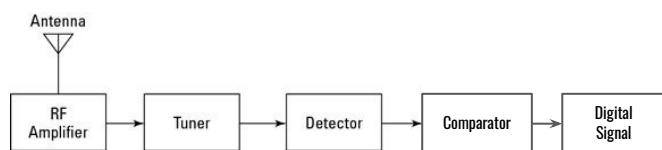
Scientific Fact and Applications

Components of a typical radio transmitter include: a **power supply** to provide electrical power to operate, an **oscillator** to create alternating current at the frequency on which the transmitter will transmit, a **modulator** to add useful information to the carrier wave either by amplitude modulation (modify wave intensity) or frequency modulation (modify wave frequency), an **amplifier** to amplify the modulated carrier wave to increase its power, and an **antenna** to convert the amplified signal to radio waves.



Simple transmitter schematic

A radio receiver typically has: an **antenna** to capture the radio waves, an **RF amplifier** that amplifies the very weak radio frequency (RF) signal from the antenna so that the signal can be processed by the tuner, a **tuner** that can extract signals of a particular frequency from a mix of signals of different frequencies, a **detector** to separate the information from the carrier wave, and a **comparator** to convert the analog signals into 1s and 0s. This may be read by a microcontroller.



Simple receiver schematic

Reference

1. <https://www.dummies.com/programming/electronics/components/radio-electronics-transmitters-and-receivers/>
2. <https://randomnerdtutorials.com/rf-433mhz-transmitter-receiver-module-with-arduino/>

Application

Telemetry is the collection of measurements or other data at remote points that uses transmitter/receiver technology. Widely used in industries such as meteorology where weather balloons transmit data and energy monitoring to measure the amount of electrical energy consumed.

Digital home automation uses this technology for functions such as remote light or switch operations.

Project

Send a simple message to another microcontroller and print the message on Serial Monitor.

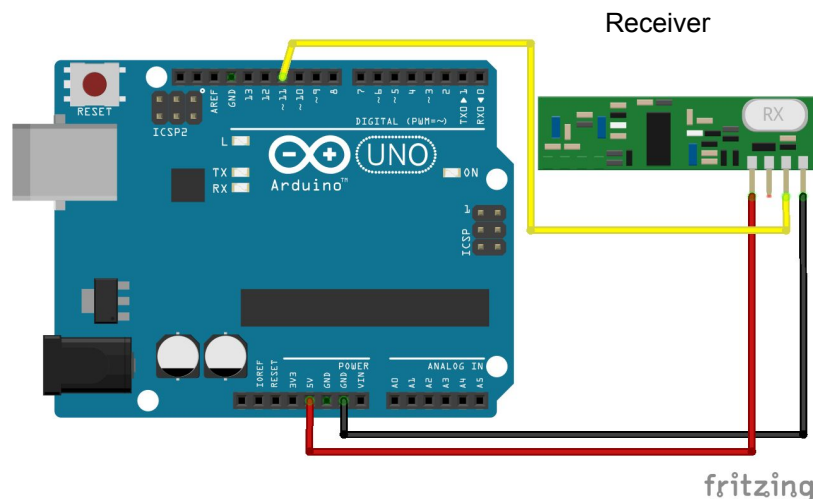
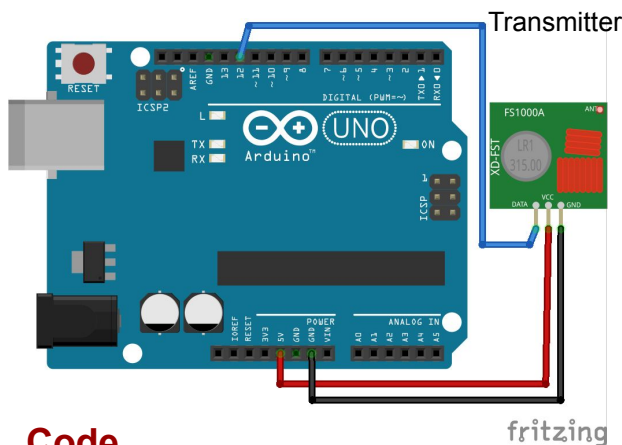
Components Required

Component	Part No.	Qty
Arduino UNO	EMX-00001-A	2
RF Transmitter and Receiver	EMC-00001-A	1

Procedure

1. install the required libraries (h files) from <https://tinyurl.com/Z2MLibraries>.
2. RF Transmitter to Arduino 1:
DATA -> Pin 12;
VCC -> 5V;
GND -> GND
3. RF Receiver to Arduino 2:
DATA -> Pin 11;
VCC -> 5V;
GND -> GND

Schematic



Code

```
/* Include these libraries*/
#include <RH_ASK.h>
#include <SPI.h>

/* Name the transmitter*/
RH_ASK driver;

void setup()
{
    /* Initiate the transmitter*/
    Serial.begin(9600);
    if (!driver.init())
        Serial.println("init failed");
}

void loop()
{
    /* Set and send message*/
    const char *msg = "Hello World!";
    driver.send((uint8_t *)msg,
        strlen(msg));
    driver.waitPacketSent();
    delay(1000);
}
```

```
/* Include these libraries*/
#include <RH_ASK.h>
#include <SPI.h> /* Not actually used but needed
to compile*/

/* Name the receiver*/
RH_ASK driver;

void setup() {
    /* Initiate the receiver*/
    Serial.begin(9600);
    if (!driver.init())
        Serial.println("init failed");
}

void loop() {
    /* Receive and print message on Serial Monitor*/
    uint8_t buf[12];
    uint8_t buflen = sizeof(buf);
    if (driver.recv(buf, &buflen)) /*Non-blocking*/
    {
        int i;
        Serial.print("Message: ");
        Serial.println((char*)buf);
    }
}
```

Challenge Yourself

1. Create a remote light switch which could control light from far by sensing noise.