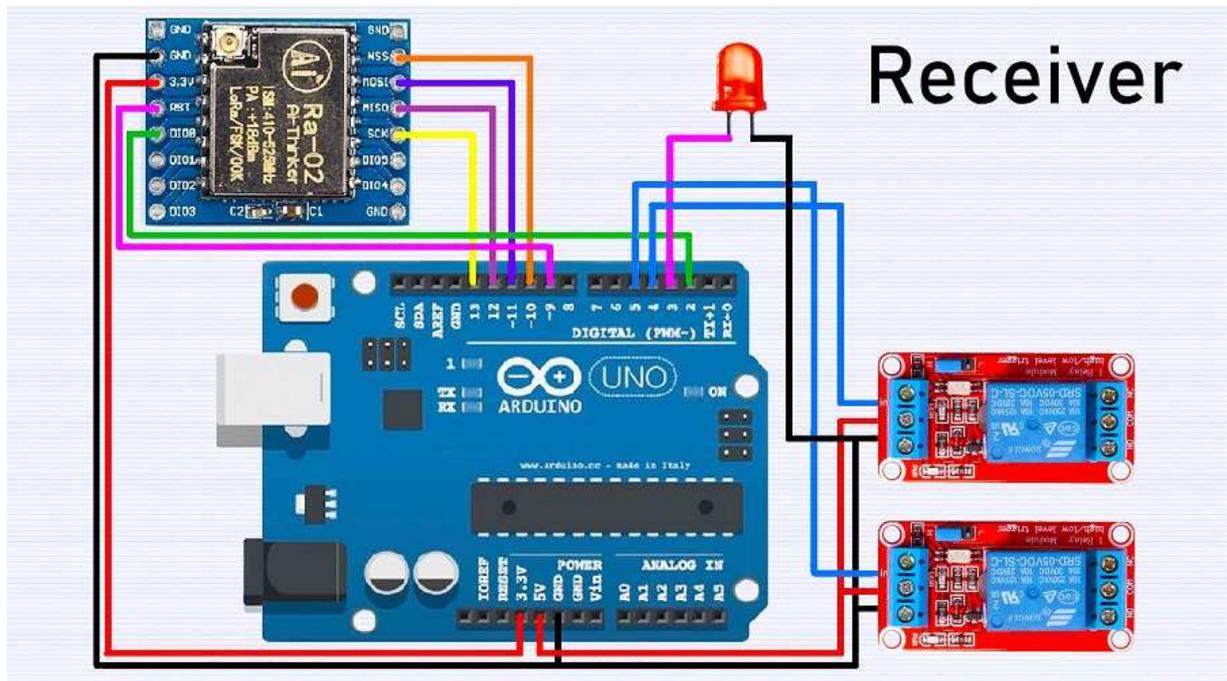


LoRa Arduino Relay Project

- 1 <https://www.youtube.com/watch?v=JKmz0MR62C8&t=197s>
- 2 https://www.youtube.com/watch?v=E4Pf_k0pJV8
- 3 https://www.youtube.com/watch?v=5_RGZ3PltE



```
//Receiver
```

```
#include <LoRa.h>
```

```
const int LED1 = 3; // indicator LED  
const int RLY1 = 4; // relay 1  
const int RLY2 = 5; // relay 2
```

```
String inString = ""; // string to hold input  
int val = 0;  
int SyncWord = 0x22;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(LED1,OUTPUT);  
  pinMode(RLY1,OUTPUT);  
  pinMode(RLY2,OUTPUT);
```

```
  digitalWrite(LED1 , LOW);  
  digitalWrite(RLY1 , HIGH);
```

```

digitalWrite(RLY2 , HIGH);

while (!Serial);
Serial.println("LoRa Receiver");
if (!LoRa.begin(433E6)) { // or 915E6
  Serial.println("Starting LoRa failed!");
  while (1);
}
LoRa.setSpreadingFactor(12); // ranges from 6-12,default 7 see API docs
LoRa.setSignalBandwidth(62.5E3); // for -139dB (page - 112)
LoRa.setCodingRate4(8); // for -139dB (page - 112)
LoRa.setSyncWord(SyncWord); // ranges from 0-0xFF, default 0x12, see API docs
/*
Serial.print("current spreading factor : ");
Serial.println(LoRa.getSpreadingFactor());
Serial.print("current bandwidth : ");
Serial.println(LoRa.getSignalBandwidth());
Serial.println("LoRa init succeeded.");
*/
}
bool i=0;
int previousValue = 0;
int liveValue = 0;

void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // read packet
    while (LoRa.available())
    {
      int inChar = LoRa.read();
      inString += (char)inChar;
      val = inString.toInt();
      digitalWrite(LED1 , HIGH);
      delay(10);
      digitalWrite(LED1 , LOW);
      delay(10);
    }
    inString = "";
    LoRa.packetRssi();
  }

  Serial.println(val);
  liveValue = val;

  if(previousValue != liveValue)
  {

```

```
previousValue = liveValue;

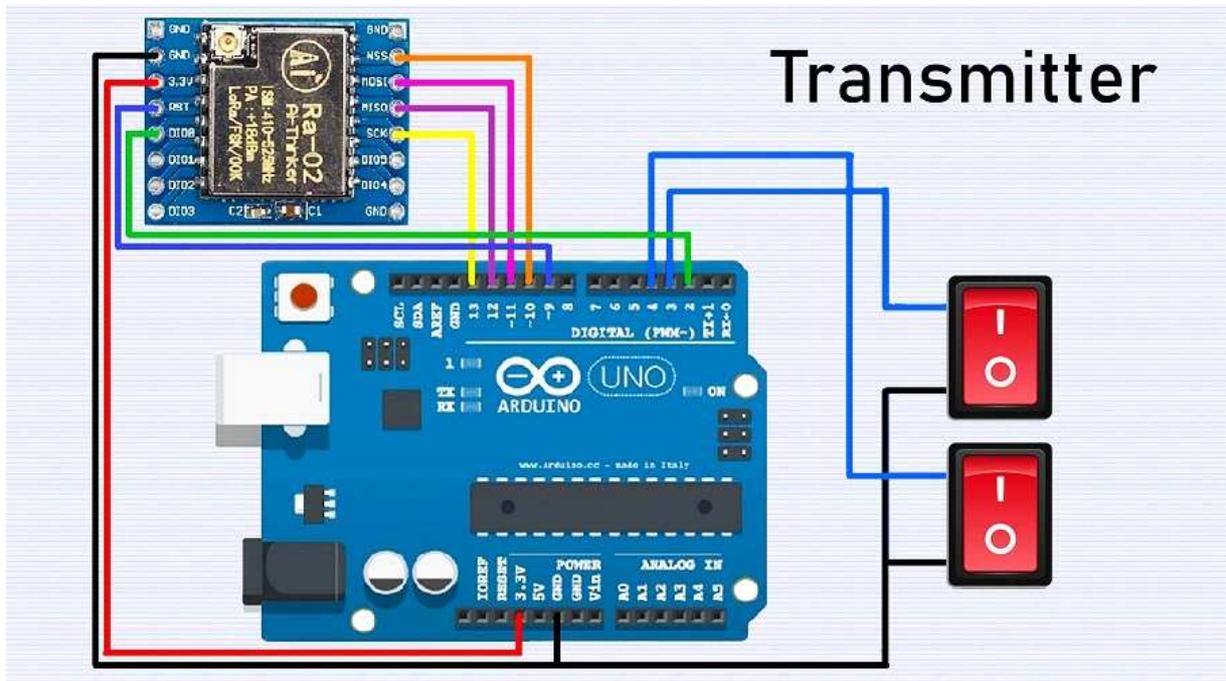
if(val == 11)
{
    digitalWrite(RLY1 , LOW);
}

if(val == 22)
{
    digitalWrite(RLY1 , HIGH);
}

if(val == 33)
{
    digitalWrite(RLY2 , LOW);
}

if(val == 44)
{
    digitalWrite(RLY2 , HIGH);
}
}
delay(50);
}

/*
* 1. when switch 1 is pressed turn ON RLY1 & RLY2
* 2. if
*/
```



//Transmitter

```
#include <LoRa.h>
//int pot = A0;
const int SW1 = 3;
const int SW2 = 4;
```

```
int SyncWord = 0x22;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(SW1,INPUT_PULLUP);
  pinMode(SW2,INPUT_PULLUP);
```

```
  cli(); //stop interrupts
  //set timer1 interrupt at 1Hz = 1sec
  TCCR1A = 0; // set entire TCCR1A register to 0
  TCCR1B = 0; // same for TCCR1B
  TCNT1 = 0; // initialize counter value to 0
  // set compare match register for 1hz increments
  OCR1A = 15624; // = (16*10^6) / (1*1024) - 1 (must be <65536)
```

```
  TCCR1B |= (1 << WGM12); // turn on CTC mode
  TCCR1B |= (1 << CS12) | (1 << CS10); // Set CS10 and CS12 bits for 1024 prescaler
  TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt
```

```
  sei(); //allow interrupts
```

```

while (!Serial);
Serial.println("LoRa Sender");
if (!LoRa.begin(433E6)) { // or 915E6, the MHz speed of your module
  Serial.println("Starting LoRa failed!");
  while (1);
}

LoRa.setSpreadingFactor(12); // ranges from 6-12, default 7 see API docs
LoRa.setSignalBandwidth(62.5E3); // for -139dB (page - 112)
LoRa.setCodingRate4(8); // for -139dB (page - 112)
LoRa.setSyncWord(SyncWord);
/*
Serial.print("current spreading factor : ");
Serial.println(LoRa.getSpreadingFactor());
Serial.print("current bandwidth : ");
Serial.println(LoRa.getSignalBandwidth());
Serial.println("LoRa init succeeded.");
*/
}

int previousSwitchValue1 = 1;
int previousSwitchValue2 = 1;
int liveSwitchValue1 = 0;
int liveSwitchValue2 = 0;
bool switchPressFlag1 = false;
bool switchPressFlag2 = false;
bool gLedPin = 0;

int data = 1;

void loop() {
  //static int data = 1;

  liveSwitchValue1 = digitalRead(SW1);
  if( (liveSwitchValue1 == 0) and (switchPressFlag1 == false) )
  {
    delay(50);
    data = 11;
    Serial.println("11");
    switchPressFlag1 = true;
    previousSwitchValue1 = !previousSwitchValue1;

    LoRa.beginPacket();
    LoRa.print(data);
    LoRa.endPacket();
  }
  if( (liveSwitchValue1 == 1) and (switchPressFlag1 == true) )

```

```

{
  delay(50);
  data = 22;
  Serial.println("22");
  switchPressFlag1 = false;
  LoRa.beginPacket();
  LoRa.print(data);
  LoRa.endPacket();
}

liveSwitchValue2 = digitalRead(SW2);
if( (liveSwitchValue2 == 0) and (switchPressFlag2 == false))
{
  delay(50);
  data = 33;
  Serial.println("33");
  switchPressFlag2 = true;
  prviousSwitchValue2 = !prviousSwitchValue2;
  LoRa.beginPacket();
  LoRa.print(data);
  LoRa.endPacket();
}
if( (liveSwitchValue2 == 1) and (switchPressFlag2 == true) )
{
  delay(50);
  data = 44;
  Serial.println("44");
  switchPressFlag2 = false;
  LoRa.beginPacket();
  LoRa.print(data);
  LoRa.endPacket();
}

if(gLedPin == 1)
{
  data = 55;
  Serial.println("55");
  gLedPin = 0;
  LoRa.beginPacket();
  LoRa.print(data);
  LoRa.endPacket();
}

// LoRa.beginPacket();
// LoRa.print(data);
// LoRa.endPacket();
}

```

```
ISR(TIMER1_COMPA_vect){//timer1 interrupt 1Hz toggles pin 13 (LED)
//generates pulse wave of frequency 1Hz/2 = 0.5kHz (takes two cycles for full wave- toggle high then
toggle low)
static volatile int ledFlag = 0;
if(++ledFlag >= 5)
{
gLedPin = 1;
ledFlag = 0;
}
}
```