

Search Projects  (/)

[Sign up \(/users/sign\\_up?redirect\\_to=%2Fkutluhan-aktar%2Firon-man-walkie-talkie-two-way-radio-for-texting-w-lora-8da393&source=nav\)](#)

Projects (/projects) Channels (/channels) News (/news) Contests (/contests) Events (/events) Videos (/videos)

**Kutluhan Aktar (/kutluhan-aktar)**Published January 13, 2022 © CC BY (<http://creativecommons.org/licenses/by/4.0>)

# Iron Man Walkie-Talkie (Two-Way Radio) for Texting w/ LoRa

Via RYLR998 LoRa module and Arduino Nano, create a two-way radio (transceiver) w/ a 4x4 keypad to transmit and receive text messages.

 Advanced(/projects?difficulty=advanced)  Full instructions provided  3 hours  
 6,036

Get all of the latest and greatest from the hardware industry in your inbox each week!



Sign up for the Hackster newsletter

([https://www.hackster.io/newsletter/sign\\_Ad](https://www.hackster.io/newsletter/sign_Ad) (<http://help.hackster.io/knowledgebase/what-are-these-ads>))

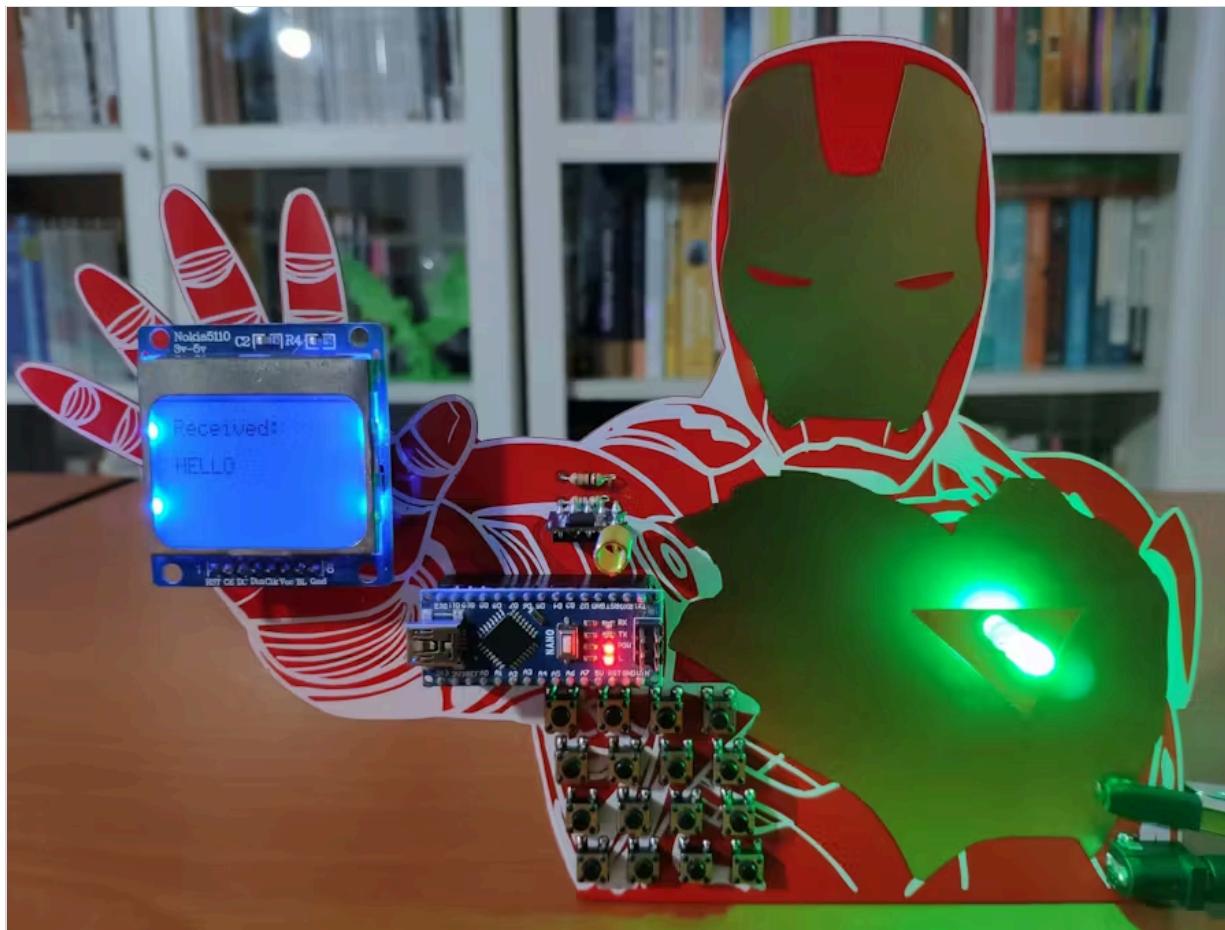
## RELATED CHANNELS AND TAGS

 (/arduino)  (/displays)

 Thanks for visiting the Hackster!  
We would love to get some feedback from you.

 KiCad (/kicad)  PCBWay (/pcbway)

 sparkfun (/sparkfun)



[display \(/projects/tags/display\)](#)

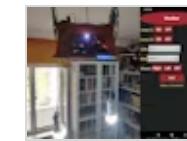
[home automation \(/projects/tags/home+automation\)](#)

[lora \(/projects/tags/lora\)](#)

[radio \(/projects/tags/radio\)](#)

[remote control \(/projects/tags/remote+control\)](#)

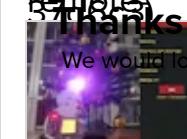
#### RELATED PROJECTS



**Bluetooth Mobile Remote Lamp with Weather Station** (/kutluhan-aktar/bluetooth-mobile-remote-lamp-with-weather-station-37d6c3)



**Joker Remote Hazardous Gas Station and Monitor w/ Arduino** (/kutluhan-aktar/joker-remote-hazardous-gas-station-and-monitor-w-arduino-62f996)



**Thanks for visiting Hackster!**  
We would love to get some feedback from you.  
**Snowman Weather and Air Quality Gift Card**  
(/kutluhan-aktar/bluetooth-enabled-snowman-weather-and-air-quality-gift-card-c87b57)

# Things used in this project

## Hardware components

PCBWay Custom PCB

(/pcbway/products/custom-  
pcb?ref=project-8da393) × 2

[View more related projects](#)

Arduino Nano R3

(/arduino/products/arduino-  
nano-r3?ref=project-  
8da393) × 2

[View more related projects](#)

Nokia 5110 Screen × 2

[View more related projects](#)

RYLR998 LoRa Module × 2

[View more related projects](#)

RGB Diffused Common  
Anode × 2

[View more related projects](#)

SparkFun Button (6x6) × 32

**Thanks for visiting Hackster!**

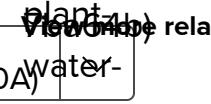
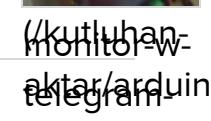
We would love to get some feedback from you.

Resistor 1k ohm × 10

[View more related projects](#)

Through Hole Resistor, 4.7  
kohm × 2

[View more related projects](#)



[View more related projects](#)

Jigglypuff IoT Carbon  
Dioxide and Dust Monitor  
w/ Telegram (/kutluhan-  
aktar/jigglypuff-iot-carbon-  
dioxide-and-dust-monitor-  
w-telegram-7ba64b)

Arduino Plant Water  
Management System w/  
BME280 (/kutluhan-  
aktar/arduino-plant-water-  
management-system-w-  
bme280-1ac1ba)

Resistor 10k ohm      × 2      [s=BAhJlhMOMzg0NDc5UHVanVjAY6BkVG%0A](https://www.hackster.io/products/pju/37908?ref=project-8da393)

Power Jack      × 2

External Battery      × 2

## Software apps and online services

Arduino IDE  
(/arduino/products/arduino-ide?ref=project-8da393)

[🔗 \(https://www.arduino.cc/en/main/software\)](https://www.arduino.cc/en/main/software)

KiCad  
(/kicad/products/kicad?ref=project-8da393)

[s=BAhJlhMOMzg0NDc5UHVanVjAY6BkVG%0A](https://www.hackster.io/products/pju/36548?ref=project-8da393)

## Hand tools and fabrication machines

Soldering iron (generic)

Solder Wire, Lead Free

[s=BAhJlhMOMzg0NDc5UHVanVjAY6BkVG%0A](https://www.hackster.io/products/pju/55952?ref=project-8da393)

Thanks for visiting Hackster!

We would love to get some feedback from you.

# Story

For a long time, I have wanted to experiment with LoRa (short for long-range), which is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. Since I am fond of vintage walkie-talkies utilizing radio waves, I decided to create a walkie-talkie (two-way radio) with a LoRa module from scratch to transmit and receive short text messages.

After perusing different LoRa module datasheets, I decided to employ the RYLR998 LoRa (transceiver) module since I wanted to make this project as compact as possible. The RYLR998 LoRa module provides ultra-long range spread spectrum communication with high sensitivity and high interference immunity while minimizing current consumption. Also, this module has a built-in antenna and supports AT commands via serial communication. You can inspect the product datasheet from here. ([https://reyax.com/wp-content/uploads/2021/07/RYLR998\\_EN.pdf](https://reyax.com/wp-content/uploads/2021/07/RYLR998_EN.pdf))

After connecting the RYLR998 module to the Arduino Nano, I utilized a Nokia 5110 screen and designed a custom 4x4 keypad with pushbuttons to obviate the need for sending and displaying text messages via the serial monitor. Finally, I added a 5mm common anode RGB LED to indicate different menu options and message notifications.

After completing wiring on a breadboard and testing the code for two LoRa-based walkie-talkies (two-way radios), I designed an Iron Man-inspired PCB for this project. I thought Iron Man was germane to my vintage walkie-talkie theme due to a scene from Iron Man 3 near the end of the movie, in which all remaining armors communicate with each other via radio waves covertly. Even

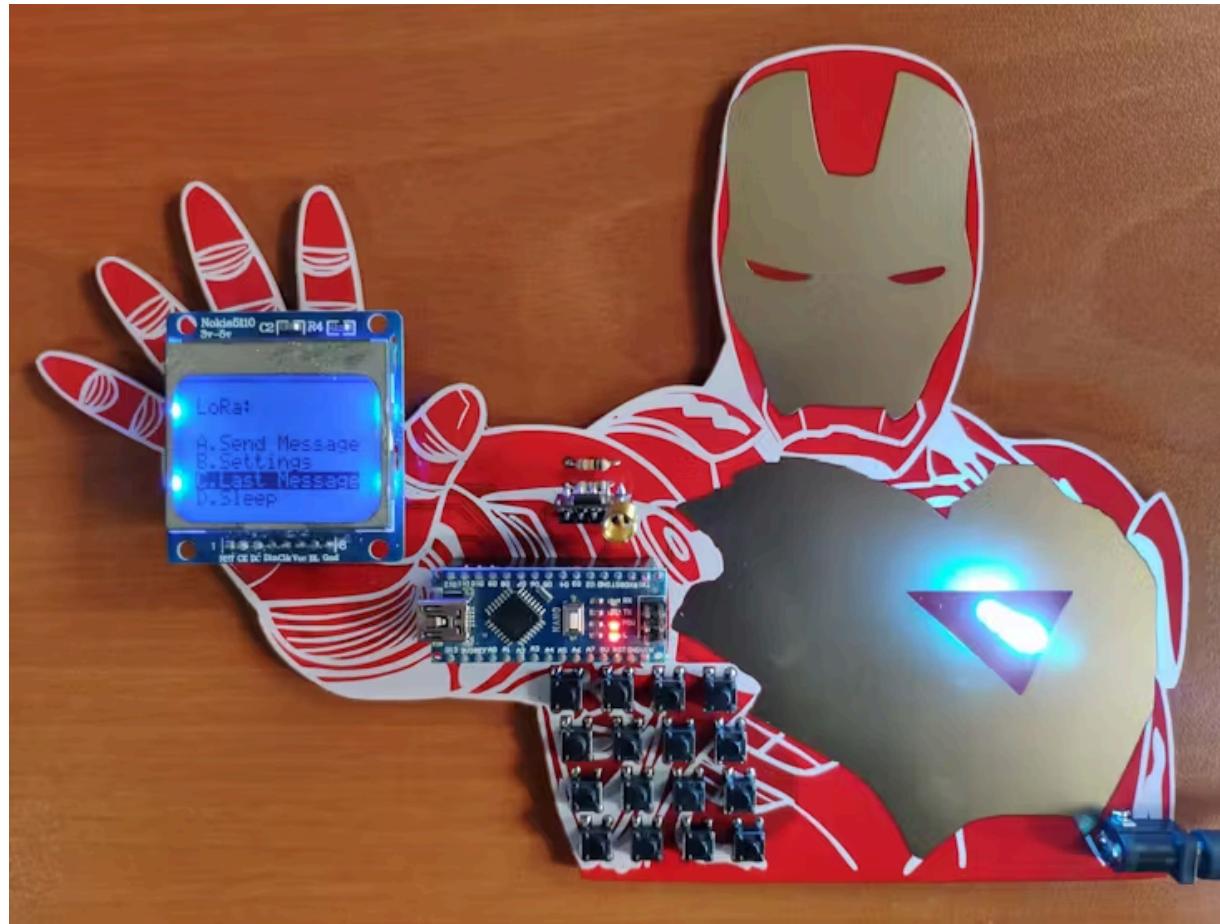
**Thanks for visiting Hackster!**

We would love to get some feedback from you.

though radio is seldom utilized by Iron Man in comics pages, I was inspired by the movie to design Iron Man-themed walkie-talkies (two-way radios) after recently rewatching it :)

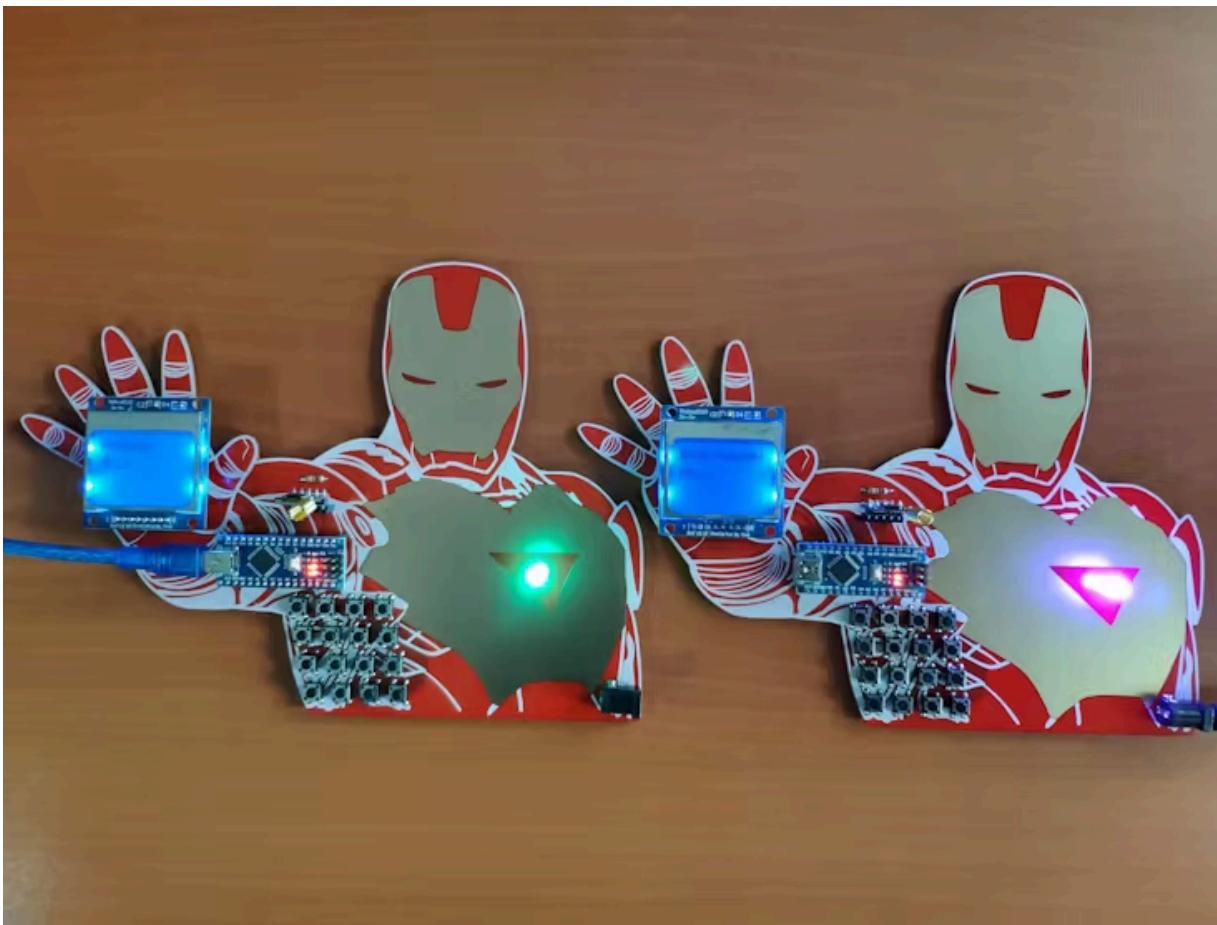
Huge thanks to PCBWay (<https://www.pcbway.com/>) for sponsoring this project.

Also, huge thanks to REYAX (<https://reyax.com/>) for sending me the RYLR998 LoRa modules. (<https://reyax.com/products/rylr998/>)



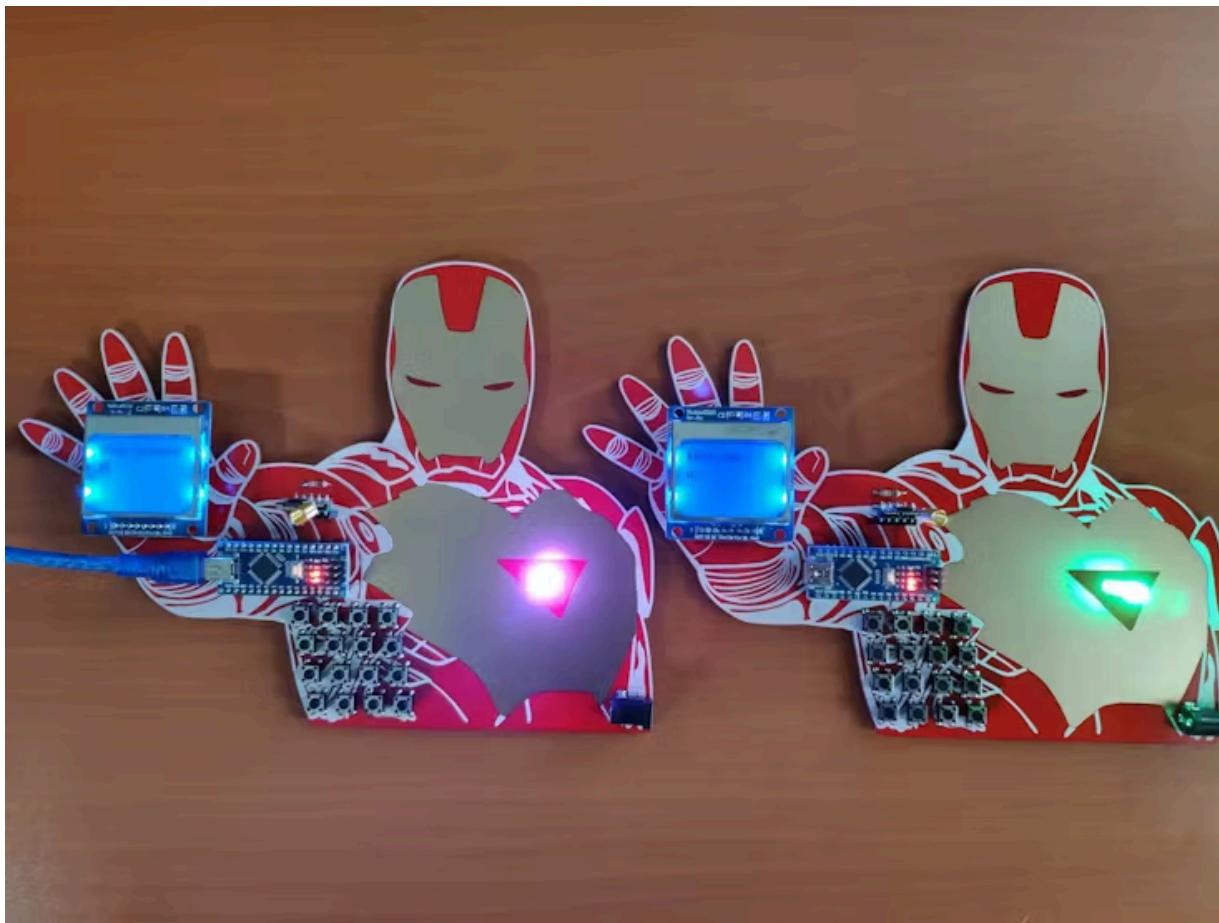
**Thanks for visiting Hackster!**

We would love to get some feedback from you.



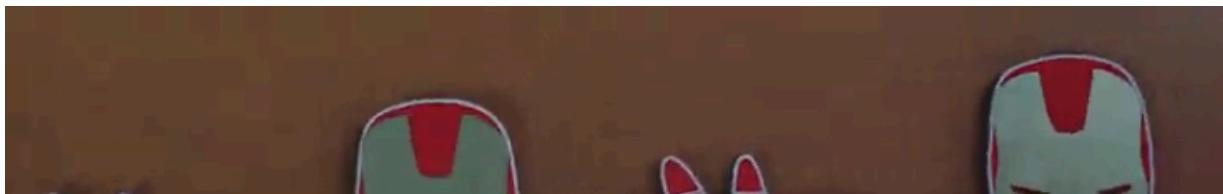
**Thanks for visiting Hackster!**

We would love to get some feedback from you.



**Thanks for visiting Hackster!**

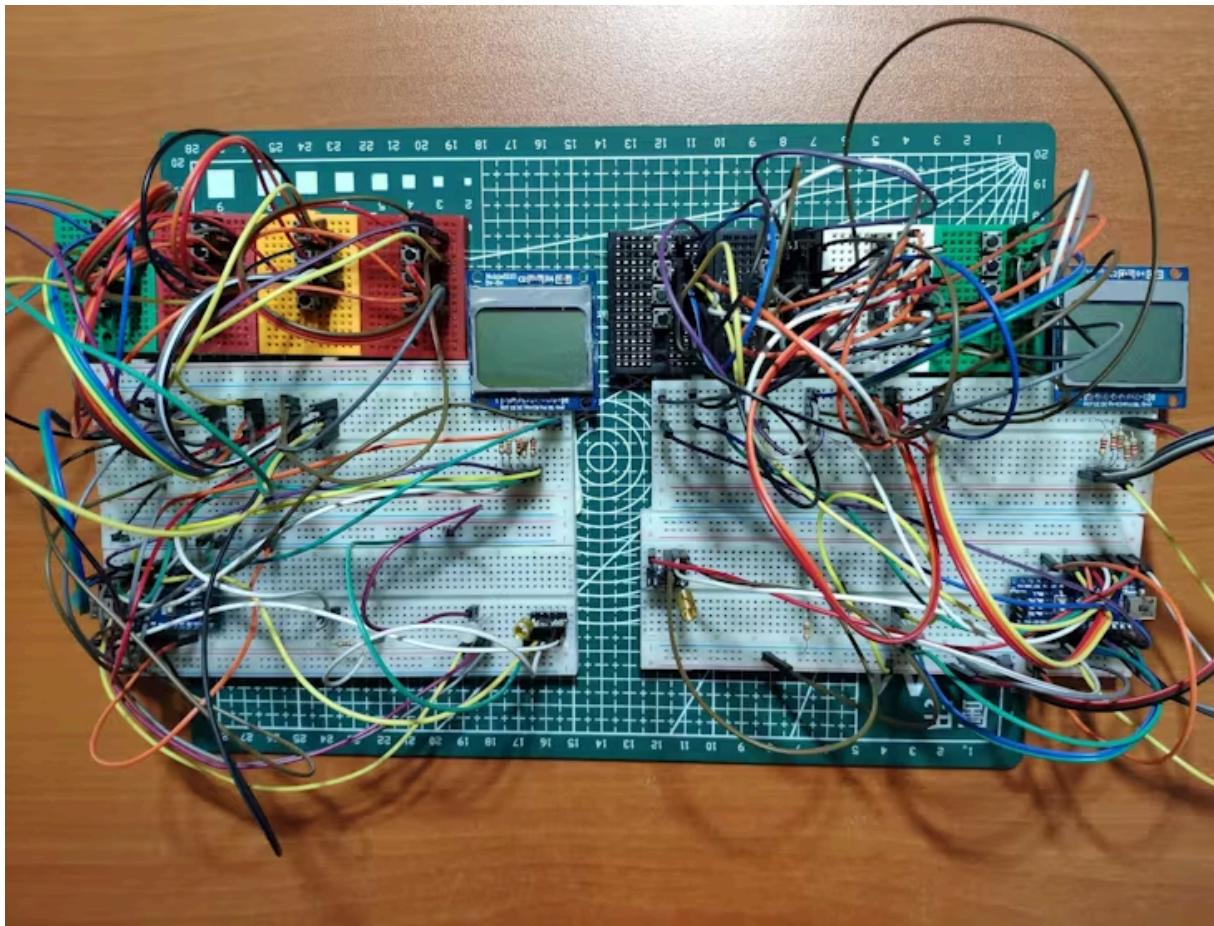
We would love to get some feedback from you.



## Step 1: Designing and soldering the Iron Man Walkie-Talkie (Two-Way Radio) PCB

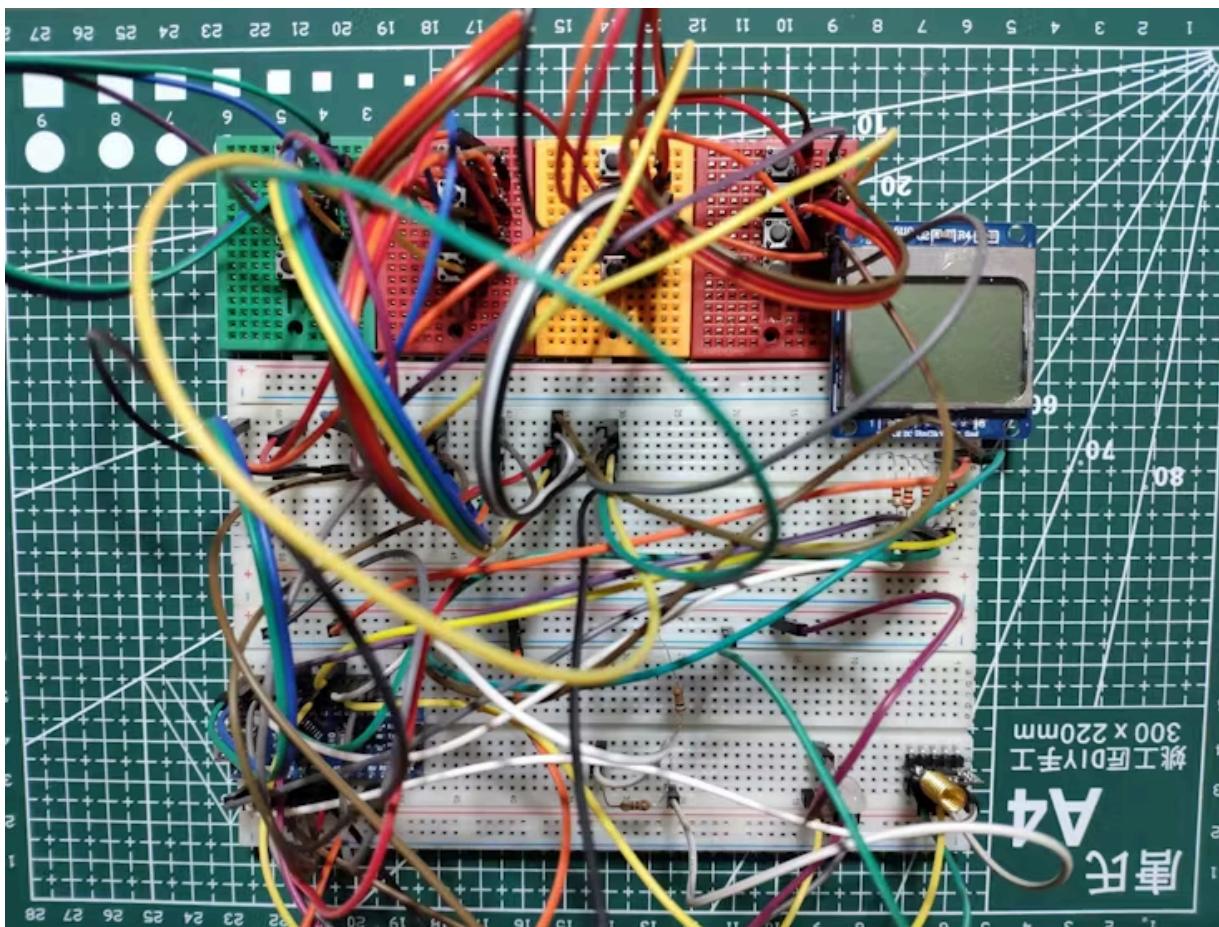
Before prototyping my Iron Man-themed PCB design, I tested all connections and wiring with the Arduino Nano and the RYLR998 LoRa module for two walkie-talkies (two-way radios).

**Thanks for visiting Hackster!**  
We would love to get some feedback from you.



**Thanks for visiting Hackster!**

We would love to get some feedback from you.



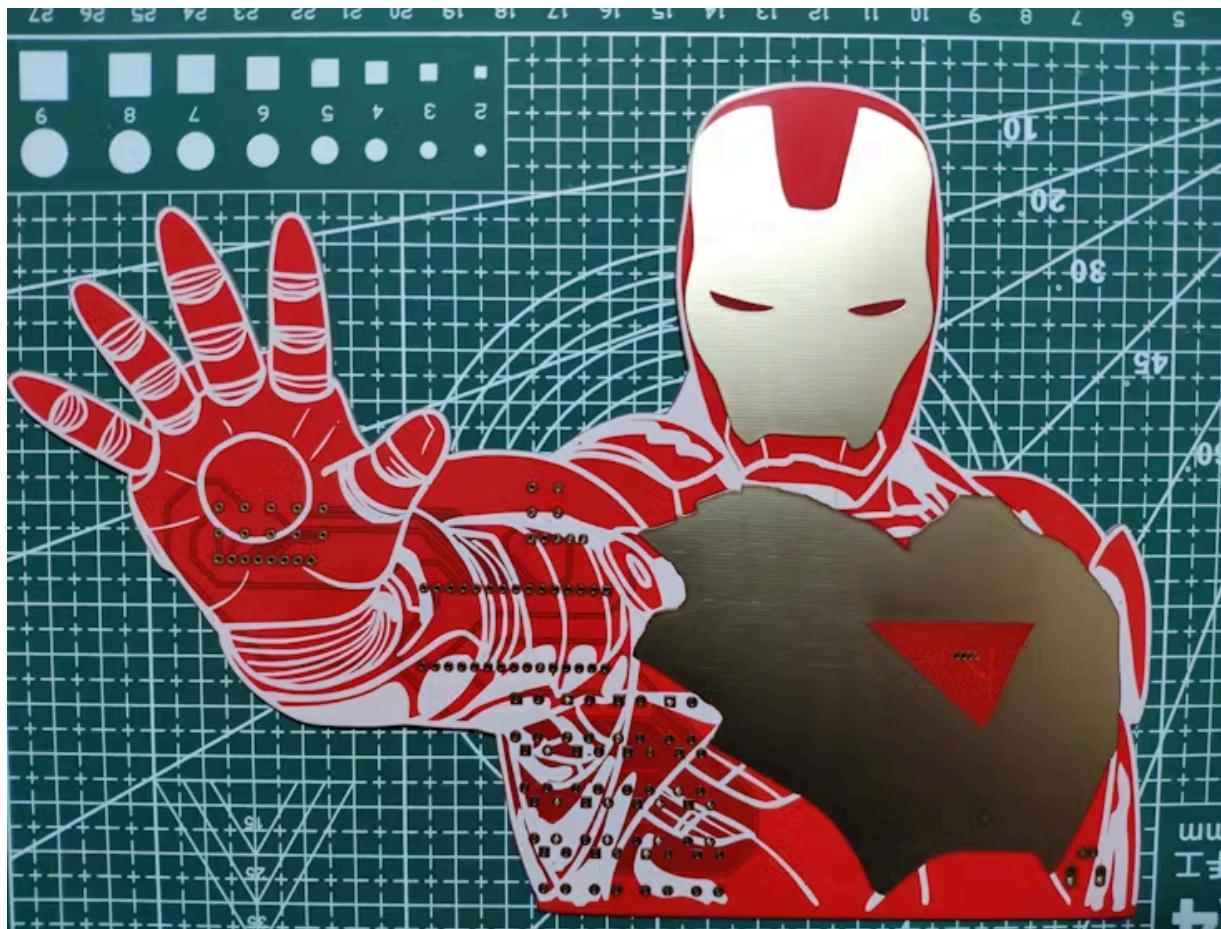
Then, I designed the Iron Man Walkie-Talkie (Two-Way Radio) PCB by utilizing KiCad - inspired by the invincible and legendary Iron Man :) I attached the Gerber file of the PCB below. Therefore, if you want, you can order this PCB from PCBWay to create your LoRa-based walkie-talkies (two-way radios) so as to transmit and receive text messages.

Click here

([https://www.pcbway.com/project/shareproject/Iron\\_Man\\_Walkie\\_Talkie\\_Two\\_Way\\_Radio\\_for\\_Texting\\_w\\_LoRa\\_02015af2.html](https://www.pcbway.com/project/shareproject/Iron_Man_Walkie_Talkie_Two_Way_Radio_for_Texting_w_LoRa_02015af2.html)) to inspect and order this PCB directly on PCBWay.

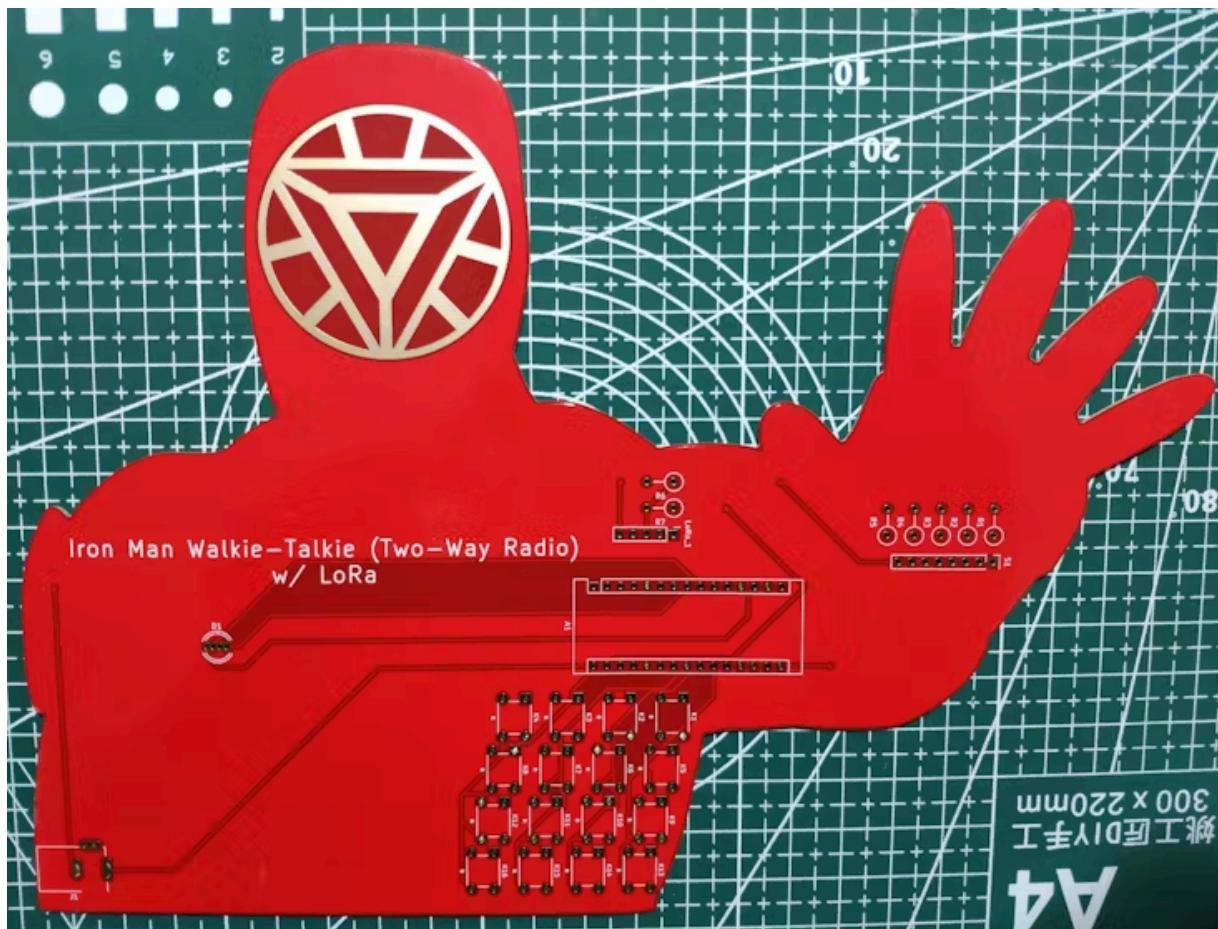
**Thanks for visiting Hackster!**

We would love to get some feedback from you.



**Thanks for visiting Hackster!**

We would love to get some feedback from you.



First of all, by utilizing a soldering iron, I attached headers (female), resistors (1K, 4.7K, 10K), pushbuttons (6x6), a 5mm common anode RGB LED, and a power jack.

Component list on the PCB:

*A1 (Headers for Arduino Nano)*

*LoRa\_1 (Headers for RYLR998 LoRa Module)*

*S1 (Headers for Nokia 5110 Screen)*

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

*D1 (5mm Common Anode RGB LED)*

*K1, K2, K3, K4, K5, K6, K7, K8, K9, K10, K11, K12, K13, K14, K15, K16 (6x6  
Pushbutton)*

*R1, R2, R3, R4, R5 (1K Resistor)*

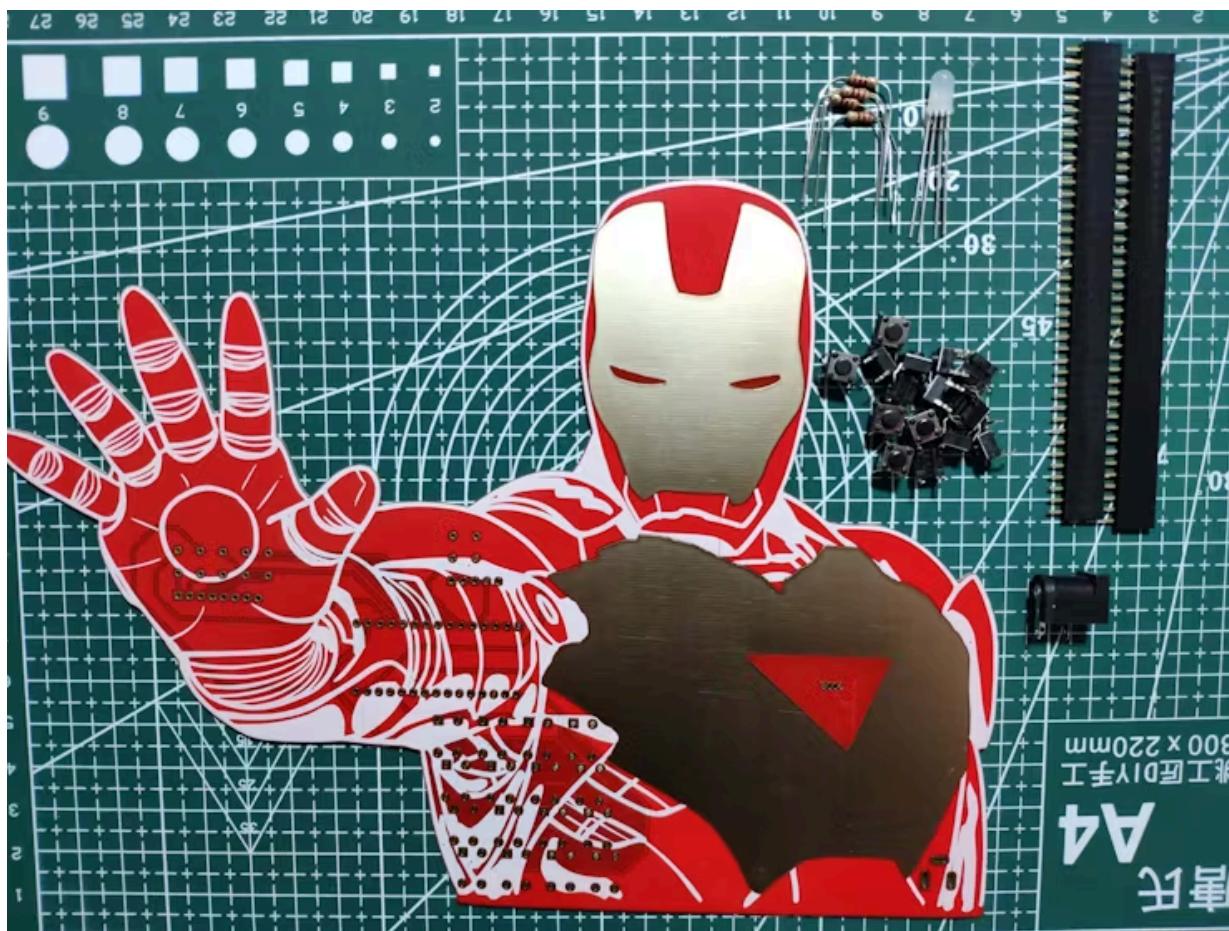
*R6 (10K Resistor)*

*R7 (4.7K Resistor)*

*J1 (Power Jack)*

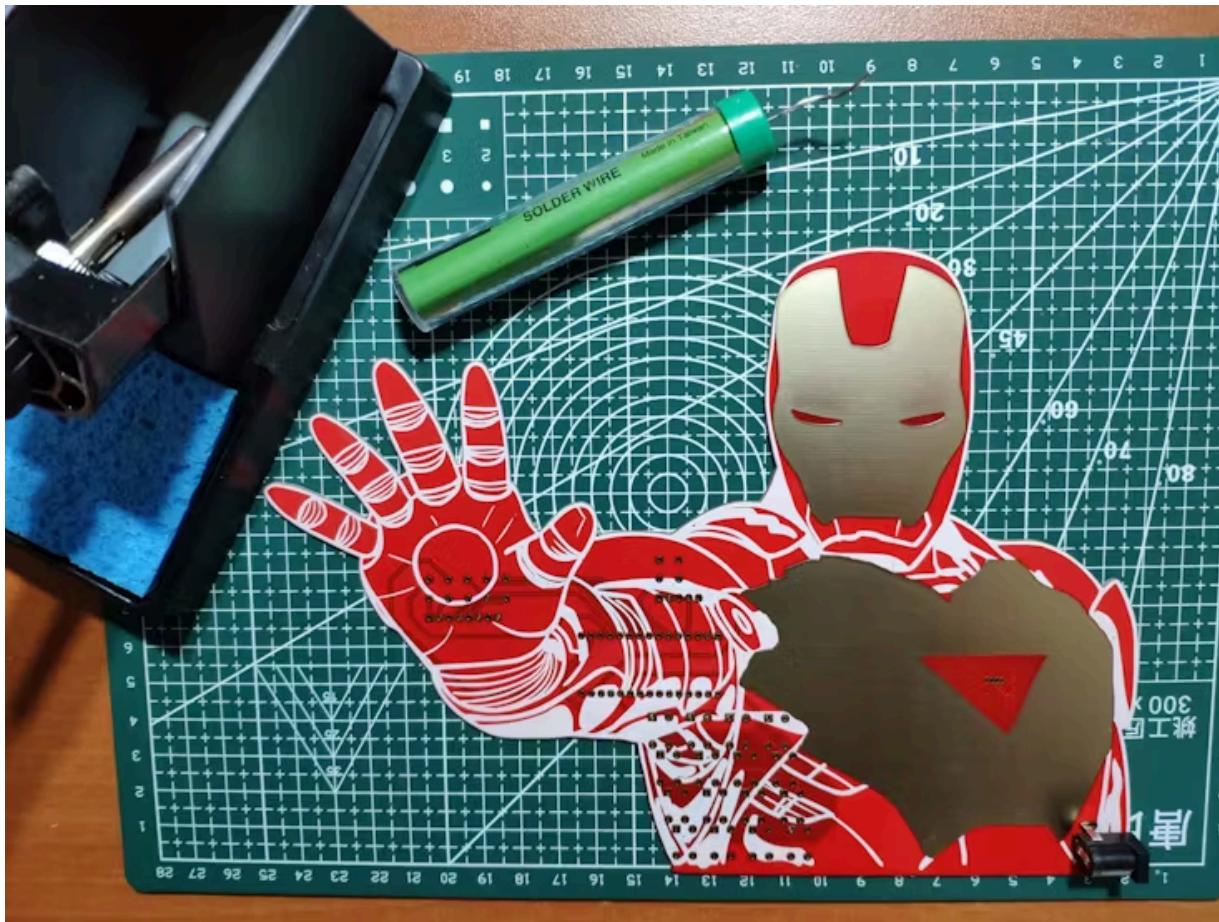
**Thanks for visiting Hackster!**

We would love to get some feedback from you.



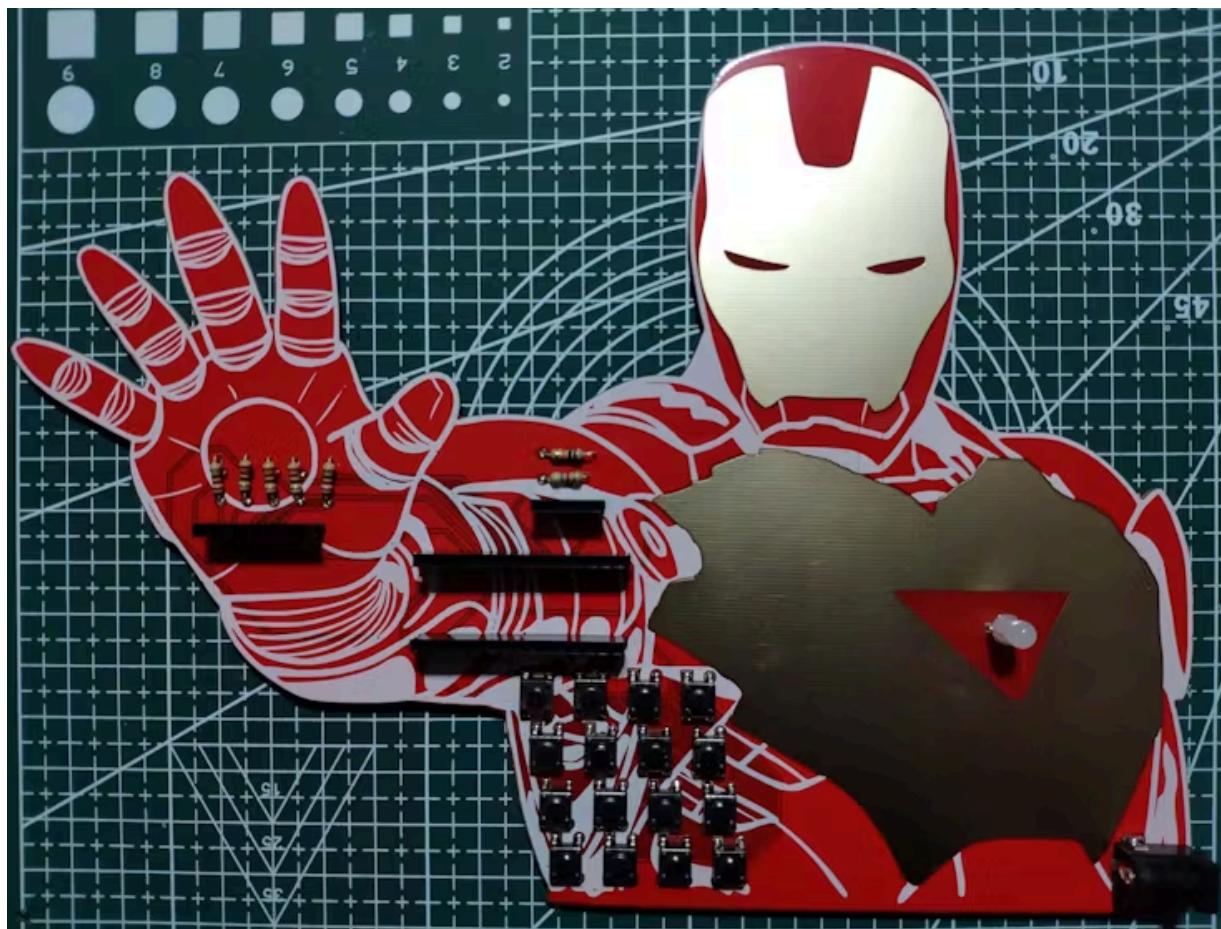
**Thanks for visiting Hackster!**

We would love to get some feedback from you.



**Thanks for visiting Hackster!**

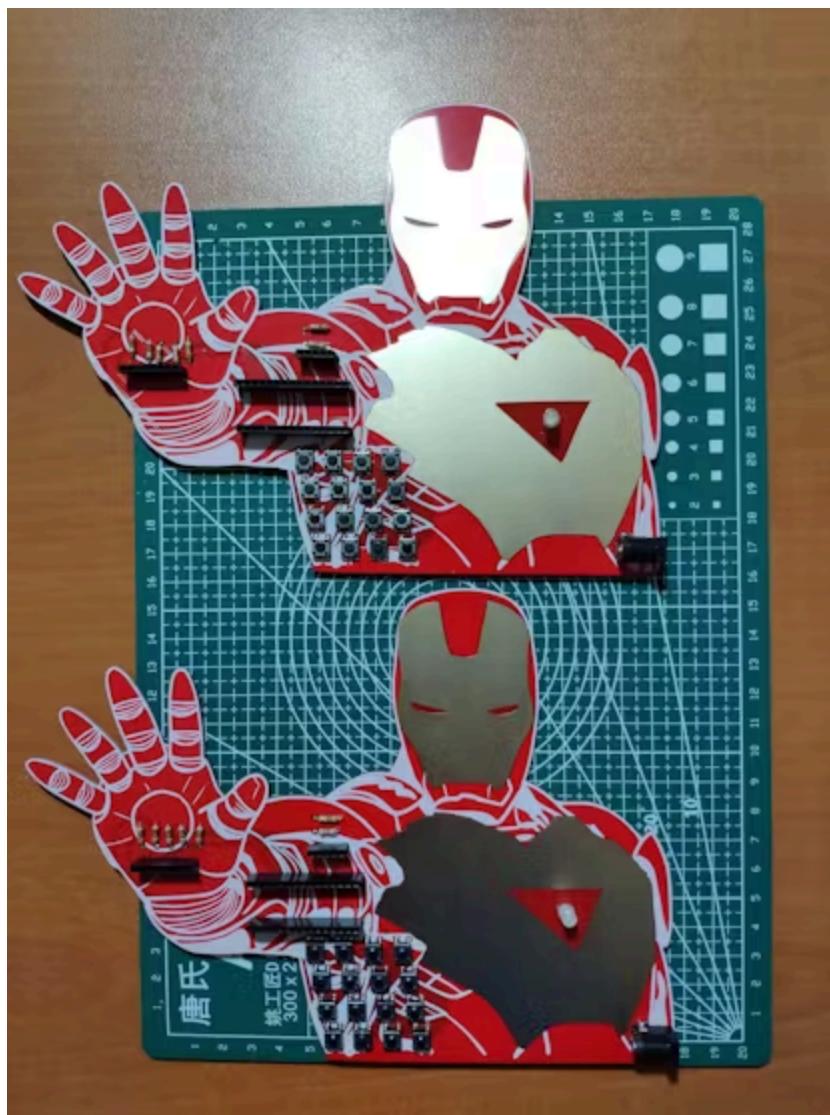
We would love to get some feedback from you.



Since I needed two Iron Man-themed walkie-talkies (two-way radios) so as to manage to transmit and receive text messages simultaneously, I soldered components to two PCBs.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.



## Step 2: Configuring and initializing the RYLR998 LoRa module

Since the RYLR998 LoRa (transceiver) module requires to be adjusted to transmit and receive text messages simultaneously, I needed to configure each RYLR998 LoRa module with specified settings and addresses.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

Fortunately, I was able to configure RYLR998 modules effortlessly with the supported AT command list via serial communication.

To configure the RYLR998 modules and send text messages, I utilized the AT commands as follows. You can inspect all available AT commands from here. ([https://reyax.com/wp-content/uploads/2021/07/Lora\\_AT\\_Command\\_RYLR998\\_RYLR498\\_EN.pdf](https://reyax.com/wp-content/uploads/2021/07/Lora_AT_Command_RYLR998_RYLR498_EN.pdf))

- # Use the **AT+ADDRESS** command to specify the RYLR998 LoRa module address while transmitting and receiving text messages.

**AT+ADDRESS=<Address>**

<Address>=0~65535 (default 0)

- # Use the **AT+NETWORKID** command to set the LoRa network ID of the RYLR998 module. Setting the correct LoRa network ID is crucial since only the RYLR998 modules with the same LoRa network ID can communicate with each other.

**AT+NETWORKID=<Network ID>**

<Network ID>=3~15, 18 (default 18)

- # Use the **AT+BAND** command to set the LoRa radio frequency (Hz) of the RYLR998 module. Two modules must have the same radio frequency to transmit and receive messages.

**AT+BAND=<parameter>**

<parameter>=

470000000: 470000000Hz

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

915000000: 915000000Hz (default)

- # Use the *AT+PARAMETER* command to set the LoRa communication parameters of the RYLR998 module. Two modules must have the same parameters to communicate with each other successfully:
- # The *spreading factor (SF)* implies the chirp rate and thus controls the speed of data transmission. Lower spreading factors mean faster chirps and higher data transmission rates. Conversely, higher spreading factors imply fewer chirps and high sensitivity with longer transmission time.
- # The *bandwidth* means channels: 125 kHz, 250 kHz, or 500 kHz. The sensitivity and the data transmission time are inversely proportional to the bandwidth.
- # The *coding rate* refers to the proportion of transmitted bits that actually carry information - 6/8, 4/8, etc.
- # The *preamble* is used to keep the receiver synchronized with the incoming data stream. The default is 12 symbol lengths.

*AT+PARAMETER=<Spreading Factor>, <Bandwidth>, <Coding Rate>, <Preamble>*

<Spreading Factor>=7~11 (default 9)

<Bandwidth>=7~9

7: 125 kHz (default)

8: 250 kHz

9: 500 kHz

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

<Coding Rate>=1^4 (default 1)

<Preamble> (default 12)

When the LoRa network ID equals 18, the *preamble* value can be configured to 4^24. Otherwise, it can only be configured to 12.

- # Use the *AT+SEND* command to transmit data (text messages) to the given RYLR998 LoRa module address. To evaluate the transmission time (airtime), employ LoRa calculator tools.

*AT+SEND=<Address>, <Payload Length>, <Data>*

<Address>=0^65535

When the receiver address is set as 0, the module transmits data to all available addresses.

<Payload Length> Maximum 240 bytes

<Data> ASCII Format

- # To send AT commands successfully to the RYLR998 module via serial communication, add \r\n to the end of all commands.

- # To obtain the current setting values of AT commands from the RYLR998 module via serial communication, add the question mark (?) to the end of them:

*AT+ADDRESS?*

*AT+SEND?*

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

To transmit and receive data (text messages) with my two RYLR998 LoRa modules successfully, I applied these settings:

📌 RYLR998 LoRa Module (1)

*AT+ADDRESS=108*

*AT+NETWORKID=10*

*AT+BAND=470000000*

*AT+PARAMETER=10, 8, 1, 12*

*AT+SEND=107, <Message Length>, <Message>*

📌 RYLR998 LoRa Module (2)

*AT+ADDRESS=107*

*AT+NETWORKID=10*

*AT+BAND=470000000*

*AT+PARAMETER=10, 8, 1, 12*

*AT+SEND=108, <Message Length>, <Message>*

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

## AT Command Set

It is required to key in “enter” or “\r\n” in the end of all AT Commands.

Add “?” in the end of the commands to ask the current setting value.

It is required to wait until the module replies +OK so that you can execute the next AT command.

### 1. AT Test if the module can respond to Commands.

Syntax	Response
AT	+OK

### 2. Software RESET

Syntax	Response
AT+RESET	+RESET +READY

### 3. AT+MODE Set the wireless work mode.

Syntax	Response
AT+MODE=<Parameter>	+OK

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

**10. AT+SEND** Send data to the appointed address by Command Mode.

- # Check the error response codes if the module cannot be configured with AT commands successfully.

**REYAX RYLR998 RYLR498 Lora AT COMMAND GUIDE** **REYAX**

**16. Error result codes**

Narrative	Response
There is not "enter" or 0xD 0xA in the end of the AT Command.	+ERR=1
The head of AT command is not "AT" string.	+ERR=2
Unknow command.	+ERR=4
The data to be sent does not match the actual length	+ERR=5
TX is over times.	+ERR=10
CRC error.	+ERR=12
TX data exceeds 240bytes.	+ERR=13
Failed to write flash memory.	+ERR=14
Unknow failure.	+ERR=15
Last TX was not completed	+ERR=17
Preamble value is not allowed.	+ERR=18
RX failed, Header error	+ERR=19
The time setting value of the "Smart receiving power saving mode" is not allowed.	+ERR=20

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

## Step 3: Programming the Arduino Nano

Download the required library to control the Nokia 5110 screen:

LCD5110\_Graph | Download (<http://www.rinkydinkelectronics.com/library.php?id=47>)

Download the required library to control the custom 4x4 keypad:

Keypad | Download (<https://www.arduino.cc/reference/en/libraries/Keypad/>)

- ★ Include the required libraries and define the RYLR998 LoRa module.

```
#include <SoftwareSerial.h>
#include <Keypad.h>
#include <LCD5110_Graph.h>

SoftwareSerial LoRa(2, 3); // RX, TX
```

- ★ Define the Nokia 5110 screen settings and the graphics (monochrome images).

★ To create different graphics (monochrome images), go to Monochrome Image Converter.

([http://www.rinkydinkelectronics.com/t\\_imageconverter\\_mono.php](http://www.rinkydinkelectronics.com/t_imageconverter_mono.php))

```
LCD5110 myGLCD(4,7,8,9,13);

extern uint8_t SmallFont[];
extern uint8_t MediumNumbers[];
// Define the graphics:
extern uint8_t iron_man[];
```

- ★ Define the symbols on the custom 4x4 keypad buttons as letters (alphabet) and numbers. Then, initialize instances of different keypad classes (k\_letters\_1, k\_letters\_2, and k\_numbers).

**Thanks for visiting Hackster!**  
We would love to get some feedback from you.

```
const byte ROWS = 4; // four rows
const byte COLS = 4; // four columns
// Define the symbols on the keypad buttons as letters (alphabet) and numbers:
char letters_1[ROWS][COLS] = {
    {'A','B','C','D'},
    {'E','F','G','H'},
    {'I','J','K','L'},
    {'#','-','*','+'}
};
char letters_2[ROWS][COLS] = {
    {'M','N','O','P'},
    {'Q','R','S','T'},
    {'U','V','W','X'},
    {'#','-','*','+'}
};
char numbers[ROWS][COLS] = {
    {'Y','Z','1','2'},
    {'3','4','5','6'},
    {'7','8','9','0'},
    {'#','-','*','+'}
};
```

★ Initiate the serial communication between the Arduino Nano and the RYLR998 LoRa module.

★ Initiate the Nokia 5110 screen.

```
LoRa.begin(115200);

// Initiate the screen.
myGLCD.InitLCD();
myGLCD.setFont(SmallFont);
```

★ In the *read\_keypad* function, obtain the recently pressed key on the 4x4 keypad and change keypad classes (k\_letters\_1, k\_letters\_2, and k\_numbers) depending on the keypad number.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

```
void read_keypad(){
    // Change keypad classes depending on the keypad number.
    if(keypad_number == 1) key = k_letters_1.getKey();
    if(keypad_number == 2) key = k_letters_2.getKey();
    if(keypad_number == 3) key = k_numbers.getKey();
}
```

★ In the *change\_menu\_options* function:

★ Increase or decrease the option number using the keypad buttons (+, -).

★ Depending on the selected option number, change the boolean status.

```
void change_menu_options(){
    // Increase or decrease the option number using the keypad buttons (+, -).
    if(key){
        if(key == '-') selected--;
        if(key == '+') selected++;
    }
    if(selected < 0) selected = 4;
    if(selected > 4) selected = 1;
    delay(100);

    // Depending on the selected option number, change the boolean status.
    switch(selected){
        case 1:
            msg_send = true; module_settings = false; last_msg = false; _sleep = false;
            break;
        case 2:
            msg_send = false; module_settings = true; last_msg = false; _sleep = false;
            break;
        case 3:
            msg_send = false; module_settings = false; last_msg = true; _sleep = false;
            break;
    }
}
```

★ In the *interface* function, display the menu options.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

```
void interface(){
    // Define options.
    myLCD.print("LoRa:", 0, 0);
    myLCD.print("A.Send Message", 0, 16);
    myLCD.print("B.Settings", 0, 24);
    myLCD.print("C.Last Message", 0, 32);
    myLCD.print("D.Sleep", 0, 40);
    myLCD.update();
}
```

- ★ In the *get\_serial\_data* function, send commands to the RYLR998 LoRa module and get responses from the module via serial communication.

```
void get_serial_data(){
    while(LoRa.available()){
        msg += (char)LoRa.read();
    }
    while(Serial.available()){
        commands += (char)Serial.read();
    }
}
```

- ★ In the *set\_AT\_commands* function, configure the RYLR998 LoRa module with the given settings via AT commands.

```
void set_AT_commands(String address, String network_ID){
    commands = "";
    LoRa.print("AT\r\n");
    delay(t);
    LoRa.print("AT+ADDRESS=" + address + "\r\n");
    delay(t);
    LoRa.print("AT+NETWORKID="+network_ID+"\r\n");
    delay(t);
    LoRa.print("AT+BAND=470000000\r\n");
    delay(t);
    LoRa.print("AT+PARAMETER=10,8,1,12\r\n");
    delay(t);
}
```

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

- ★ In the *check\_AT\_settings* function, obtain the current setting values of AT commands from the RYLR998 module.

```
void check_AT_settings(){
    commands = "";
    LoRa.print("AT\r\n");
    delay(t);
    LoRa.print("AT+ADDRESS?\r\n");
    delay(t);
    LoRa.print("AT+NETWORKID?\r\n");
    delay(t);
    LoRa.print("AT+BAND?\r\n");
    delay(t);
    LoRa.print("AT+PARAMETER?\r\n");
    delay(t);
}
```

- ★ In the *scrolling\_text* function, scroll the given text by using the '#' keypad button.

```
void scrolling_text(String text, int y){
    int len = text.length();
    // Scroll the given text using the keypad button (#).
    if(key == '#') x--;
    if(x<=(len*6)) x = -(len*6);
    // Print.
    myLCD.print(text, x, y);
    delay(25);
}
```

- ★ If there are incoming bytes from the transmitter RYLR998 LoRa module, elicit and format the received message by utilizing the given delimiter - the percent sign (%).

+RCV=108, 7, %HELLO%, -7, 11

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

★ Then, blink the RGB LED as green and print the recently received message on the screen.

★ Scroll the received message by pressing the '#' keypad button.

★ If the '-' keypad button is pressed, return to the interface.

```
if(msg != ""){
    Serial.print(msg);
    if(msg.indexOf("RCV") > 0){
        myLCD.clrScr();
        myLCD.update();
        activated = true;
        // Elicit and format the received message:
        int delimiter_1, delimiter_2;
        delimiter_1 = msg.indexOf("%");
        delimiter_2 = msg.indexOf("%", delimiter_1 + 1);
        // Glean information as substrings.
        received_msg = msg.substring(delimiter_1 + 1, delimiter_2);
        adjustColor(0,255,0); delay(1000); adjustColor(0,0,0); delay(1000); adjustColor(0,255,0);
        while(activated == true){
            read_keypad();
            myLCD.print("Received:", 0, 0);
            scrolling_text(received_msg, 16);
            myLCD.update();
            // Exit.
            if(key && key == '-'){ activated = false; msg = ""; received_msg = ""; x = 0; }
        }
    }
}
```

★ After being selected, if the *Send Message* menu option is activated by pressing the '\*' keypad button:

★ Change the keypad classes by pressing the '+' keypad button:

- *k\_letters\_1*
- *k\_letters\_2*

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

- *k\_numbers*

★ Enter the text message.

★ Scroll the entered message by pressing the '#' keypad button.

★ Press the '\*' keypad button so as to transmit the recently entered text message to the receiver RYLR998 LoRa module.

★ If the given text message is transmitted successfully, blink the RGB LED as yellow.

★ If the '-' keypad button is pressed, return to the interface.

```
if(msg_send){  
    do{  
        myLCD.invertText(true);  
        myLCD.print("A.Send Message", 0, 16);  
        myLCD.invertText(false);  
        myLCD.update();  
        adjustColor(255, 0, 255);  
        delay(100);  
        if(key && key == '*'){  
            myLCD.clrScr();  
            myLCD.update();  
            activated = true;  
            while(activated == true){  
                read_keypad();  
                get_serial_data();  
                myLCD.print("Enter Message:", 0, 0);  
                if(key){  
                    if(key == '*' && MESSAGE != ""){  
                        // Send the recently entered message to the given RYLR998 transceive  
                        r module.  
                        LoRa.print("AT+SEND=107,"+String(MESSAGE.length()+2)+","%"+MESSAGE+"%"
```

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

★ After being selected, if the *Settings* menu option is activated by pressing the '\*' keypad button:

★ Utilizing the serial monitor, check and modify the RYLR998 LoRa module settings with AT commands.

★ If the '-' keypad button is pressed, return to the interface.

```
if(module_settings){  
    do{  
        myLCD.invertText(true);  
        myLCD.print("B.Settings", 0, 24);  
        myLCD.invertText(false);  
        myLCD.update();  
        adjustColor(0, 0, 255);  
        delay(100);  
        if(key && key == '*'){  
            myLCD.clrScr();  
            myLCD.update();  
            activated = true;  
            while(activated == true){  
                read_keypad();  
                get_serial_data();  
                myLCD.print("Use the serial", 0, 0);  
                myLCD.print("monitor to", 0, 8);  
                myLCD.print("check and set", 0, 16);  
                myLCD.print("module", 0, 24);  
                myLCD.print("settings!", 0, 32);  
                // Utilizing the serial monitor, check and set the RYLR998 module setting  
            }  
        }  
    }  
}
```

★ After being selected, if the *Last Message* menu option is activated by pressing the '\*' keypad button:

★ If the '\*' keypad button is pressed, obtain and display the latest transmitted message by utilizing the given delimiter - the percent sign (%).

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

- ★ If there is no previously transmitted message saved in the module's memory, then print *Nothing Transmitted Yet!* as the latest message.
- ★ Scroll the latest transmitted message by pressing the '#' keypad button.
- ★ If the '-' keypad button is pressed, return to the interface.

```
if(last_msg){  
    do{  
        myLCD.invertText(true);  
        myLCD.print("C.Last Message", 0, 32);  
        myLCD.invertText(false);  
        myLCD.update();  
        adjustColor(0, 255, 255);  
        delay(100);  
        if(key && key == '*'){  
            myLCD.clrScr();  
            myLCD.update();  
            activated = true;  
            while(activated == true){  
                read_keypad();  
                get_serial_data();  
                myLCD.print("Last Message:", 0, 0);  
                // Obtain and display the latest transmitted message.  
                if(key && key == '*') LoRa.print("AT+SEND?\r\n");  
                if(msg != ""){  
                    // Print the response from the RYLR998 module.  
                    Serial.println(msg);  
                }  
            }  
        }  
    }  
}
```

- ★ After being selected, if the *Sleep* menu option is activated by pressing the '\*' keypad button:
  - ★ Display the monochrome image (*iron\_man*) as the screensaver while sleeping.
  - ★ If the '-' keypad button is pressed, return to the interface.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

```
if(_sleep){  
    do{  
        myLCD.invertText(true);  
        myLCD.print("D.Sleep", 0, 40);  
        myLCD.invertText(false);  
        myLCD.update();  
        adjustColor(255, 0, 0);  
        delay(100);  
        if(key && key == '*'){  
            myLCD.clrScr();  
            myLCD.update();  
            activated = true;  
            while(activated == true){  
                read_keypad();  
                // Define and print monochrome images on the screen:  
                myLCD.drawBitmap(25,0,iron_man,36,50);  
                myLCD.update();  
                // Exit.  
                if(key && key == '-'){ activated = false; myLCD.clrScr(); myLCD.update()  
            }; }  
    }  
}
```

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

```
Iron_Man_Walkie-Talkie_w_LoRa iron_man.c
```

// Include the required libraries.

```
Iron_Man_Walkie-Talkie_w_LoRa iron_man.c
```

```
// Initialize instances of different keypad classes:  
Keypad k_letters_1 = Keypad( makeKeymap(letters_1), rowPins, colPins, ROWS, COLS);  
Keypad k_letters_2 = Keypad( makeKeymap(letters_2), rowPins, colPins, ROWS, COLS);  
Keypad k_numbers = Keypad( makeKeymap(numbers), rowPins, colPins, ROWS, COLS);  
  
// Define RGB pins:  
#define redPin 5  
#define greenPin 6  
#define bluePin 10  
  
// Define interface options:  
volatile boolean msg_send, module_settings, last_msg, _sleep, activated;  
  
// Define the data holders:  
char key;  
String commands = "", msg = "", MESSAGE = "", latest_message = "", received_msg = "";  
int t = 150, selected = 0, keypad_number = 1, x = 0;  
  
void setup(){  
    Serial.begin(115200);  
    while (!Serial) {  
        : // Wait for the serial port to connect. Needed for the Native USB only.  
    }  
  
    // RGB:  
    pinMode(redPin, OUTPUT);  
    pinMode(greenPin, OUTPUT);  
    pinMode(bluePin, OUTPUT);  
    addInvertColor(0 0 0);
```

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

[Iron\\_Man\\_Walkie-Talkie\\_w\\_LoRa](#) [iron\\_man.c](#)

```
Iron_Man_Walkie-Talkie_w_LoRa iron_man.c

if(msg_send){
do{
    myLCD.invertText(true);
    myLCD.print("A.Send Message", 0, 16);
    myLCD.invertText(false);
    myLCD.update();
    adjustColor(255, 0, 255);
    delay(100);
    if(key == '*'){
        myLCD.clrScr();
        myLCD.update();
        activated = true;
        while(activated == true){
            read_keypad();
            get_serial_data();
            myLCD.print("Enter Message:", 0, 0);
            if(key){
                if(key == '*' && MESSAGE != ""){
                    // Send the recently entered message to the given RYLR998 transceiver module.
                    LoRa.print("AT+SEND=107,"+String(MESSAGE.length()+2)+",""+MESSAGE+"\r\n");
                    delay(t);
                    adjustColor(255, 255, 0); delay(1000); adjustColor(255, 0, 255);
                }else if(key == '#'){
                    // Change keypad classes:
                    keypad_number++;
                    if(keypad_number > 3) keypad_number = 1;
                }else{
                    if(key != '#') MESSAGE += key;
                    scrolling_text(MESSAGE, 16);
                }
            }
        }
    }
}
```

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

```
iron_Man_Walkie-Talkie_w_LoRa iron_man.c

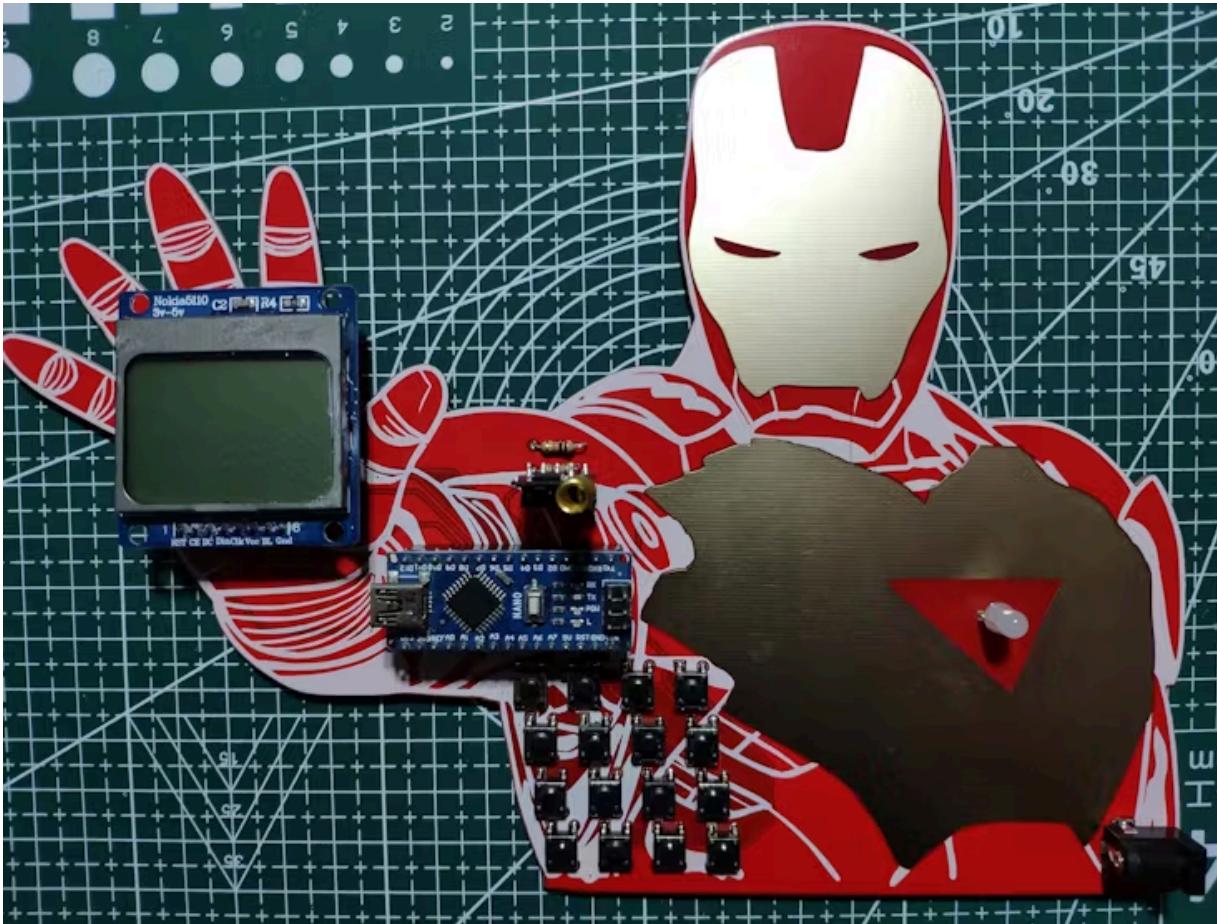
void read_keypad() {
    // Change keypad classes depending on the keypad number.
    // D4      ----- SCK (Clk)
    // D7      ----- MOSI (Din)
    // D8      ----- DC
    // D9      ----- RST
    // D13     ----- CS (CE)
    //          ----- RYLR998 LoRa Module
    // D2      ----- TX
    // D3      ----- RX
    //          ----- RGB LEB (RAGB)
    // D5      ----- R
    // D6      ----- G
    // D10     ----- B
    //          ----- 4x4 Keypad
    // D11     ----- R1
    // D12     ----- R2
    // A0      ----- R3
    // A1      ----- R4
    // A2      ----- C1
```

## Connections and Adjustments

```
// Connections
// Arduino Nano :
//          ----- Nokia 5110 Screen
// D4      ----- SCK (Clk)
// D7      ----- MOSI (Din)
// D8      ----- DC
// D9      ----- RST
// D13     ----- CS (CE)
//          ----- RYLR998 LoRa Module
// D2      ----- TX
// D3      ----- RX
//          ----- RGB LEB (RAGB)
// D5      ----- R
// D6      ----- G
// D10     ----- B
//          ----- 4x4 Keypad
// D11     ----- R1
// D12     ----- R2
// A0      ----- R3
// A1      ----- R4
// A2      ----- C1
```

After completing soldering and uploading the code, I attached all remaining components to the Iron Man-themed PCBs via headers - Arduino Nano boards, Nokia 5110 screens, and RYLR998 LoRa modules.

**Thanks for visiting Hackster!**  
We would love to get some feedback from you.



## Modes and Features

📞 The device shows four different modes (menu options) on the interface:

- A. Send Message
- B. Settings
- C. Last Message
- D. Sleep

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

 The device allows the user to select a mode (option) on the interface

via the '-', '+', and '\*' keypad buttons:

- '-' → Go Up
- '+' → Go Down
- '\*' → Activate

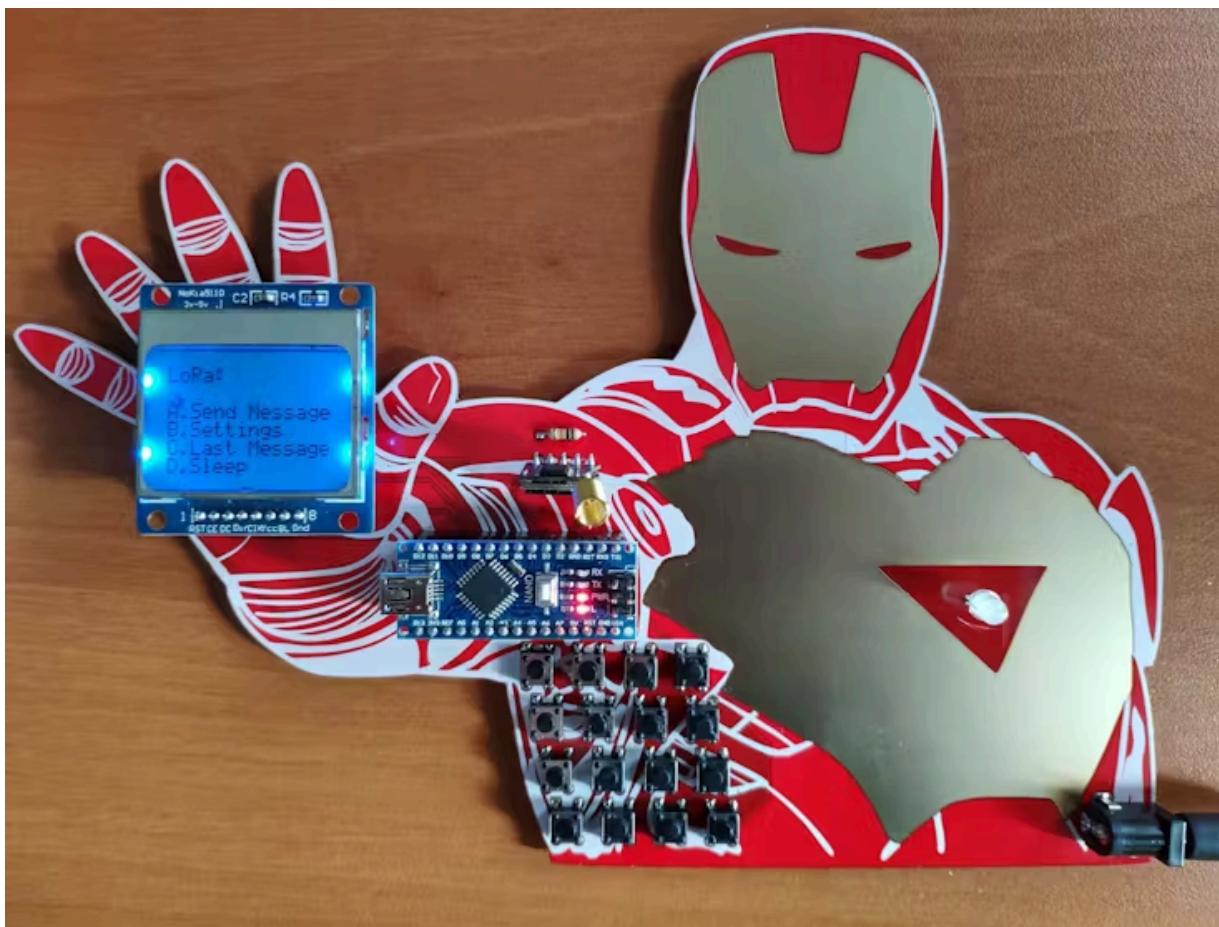
 While selecting among modes on the interface, the device turns the

RGB LED to different colors for each mode:

- Send Message → Purple
- Settings → Blue
- Last Message → Cyan
- Sleep → Red

**Thanks for visiting Hackster!**

We would love to get some feedback from you.



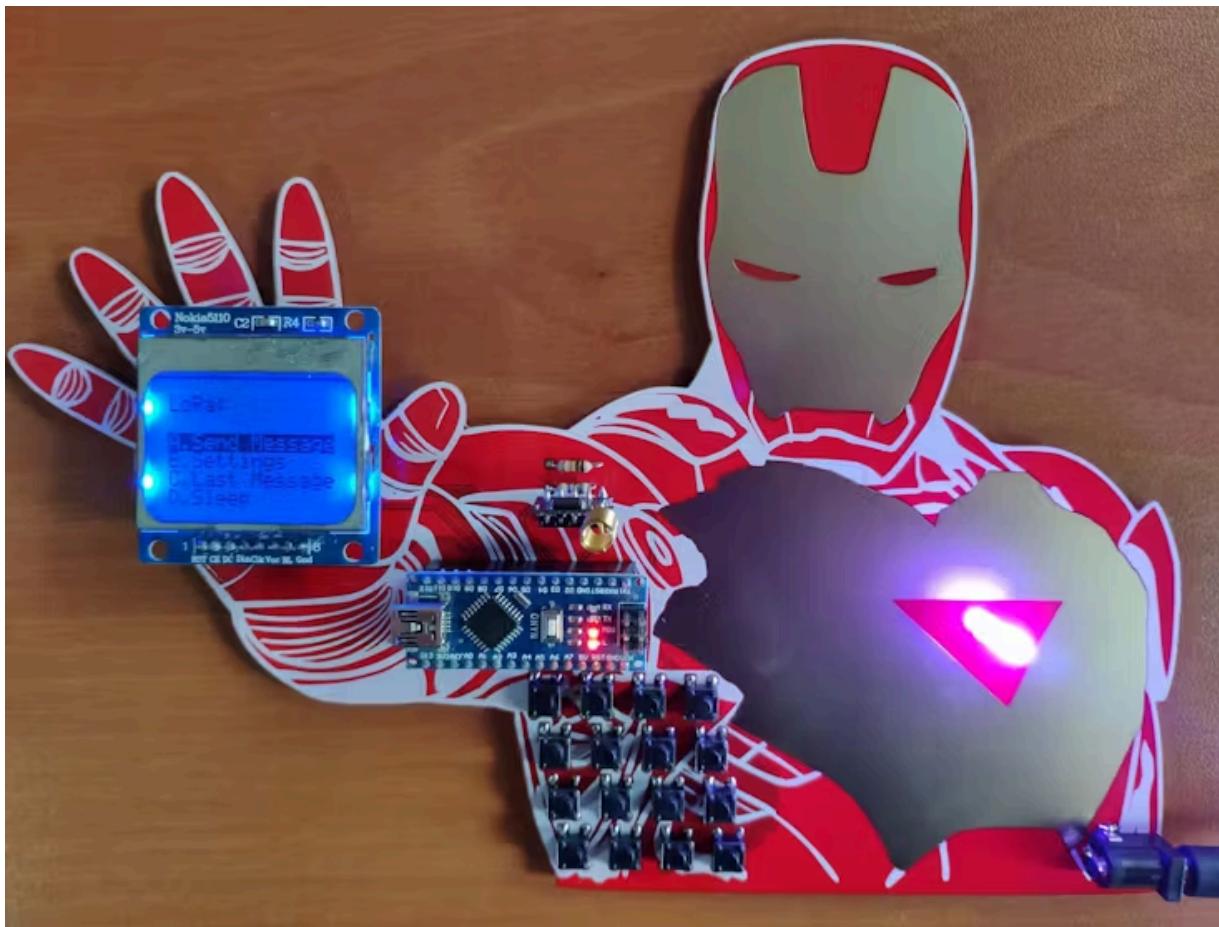
📞 After activating any modes, the device lets the user return to the interface by pressing the '-' keypad button.

📌 Modes (Menu Options):

📌 A. Send Message

**Thanks for visiting Hackster!**

We would love to get some feedback from you.



📞 The device changes the 4x4 keypad pattern (symbols) with three different keymaps if the '+' keypad button is pressed:

💻 letters\_1

'A', 'B', 'C', 'D'

'E', 'F', 'G', 'H'

'I', 'J', 'K', 'L'

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

'#', '‐', '‐\*', '‐+'

## letters\_2

'M', 'N', 'O', 'P'

'Q', 'R', 'S', 'T'

'U', 'V', 'W', 'X'

'#', '‐', '‐\*', '‐+'

## numbers

'Y', 'Z', '1', '2'

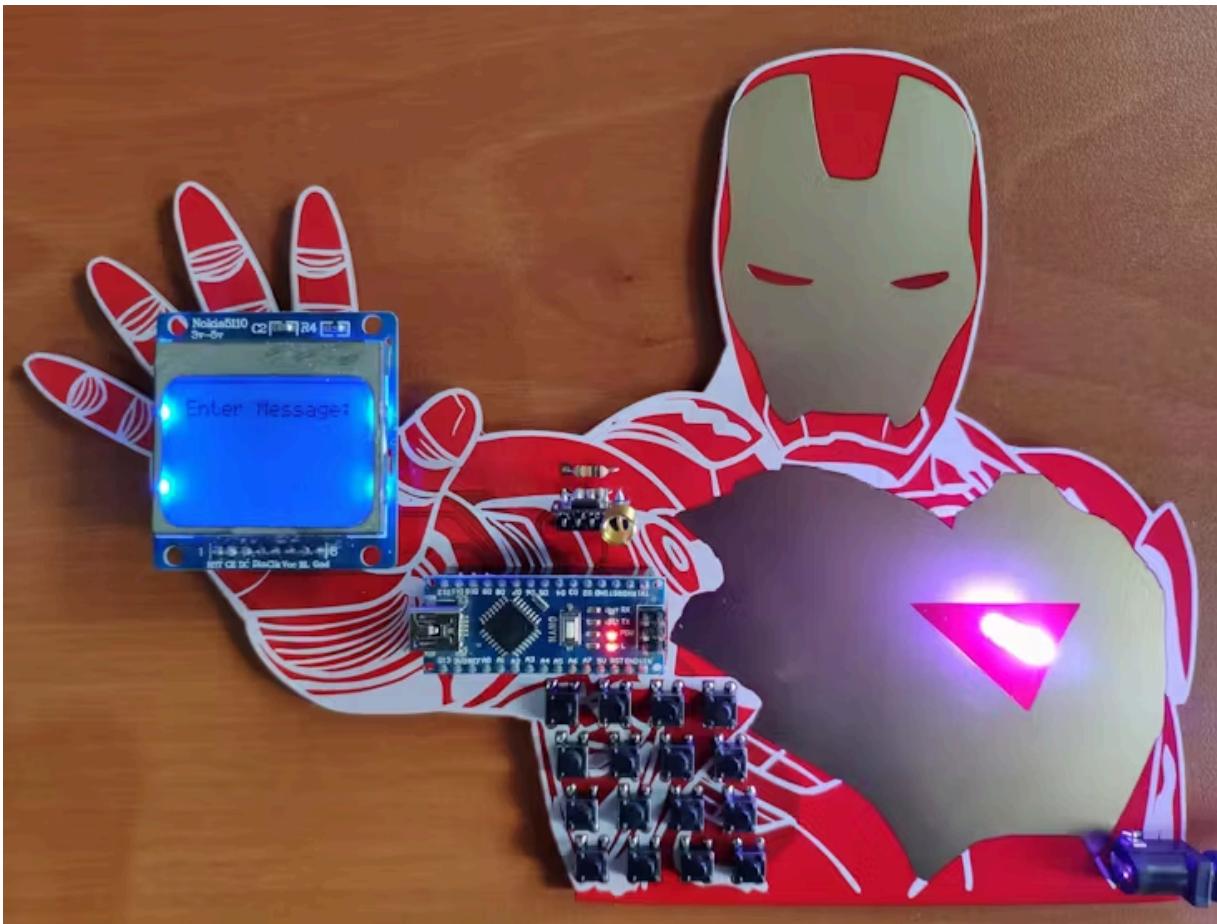
'3', '4', '5', '6'

'7', '8', '9', '0'

'#', '‐', '‐\*', '‐+'

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

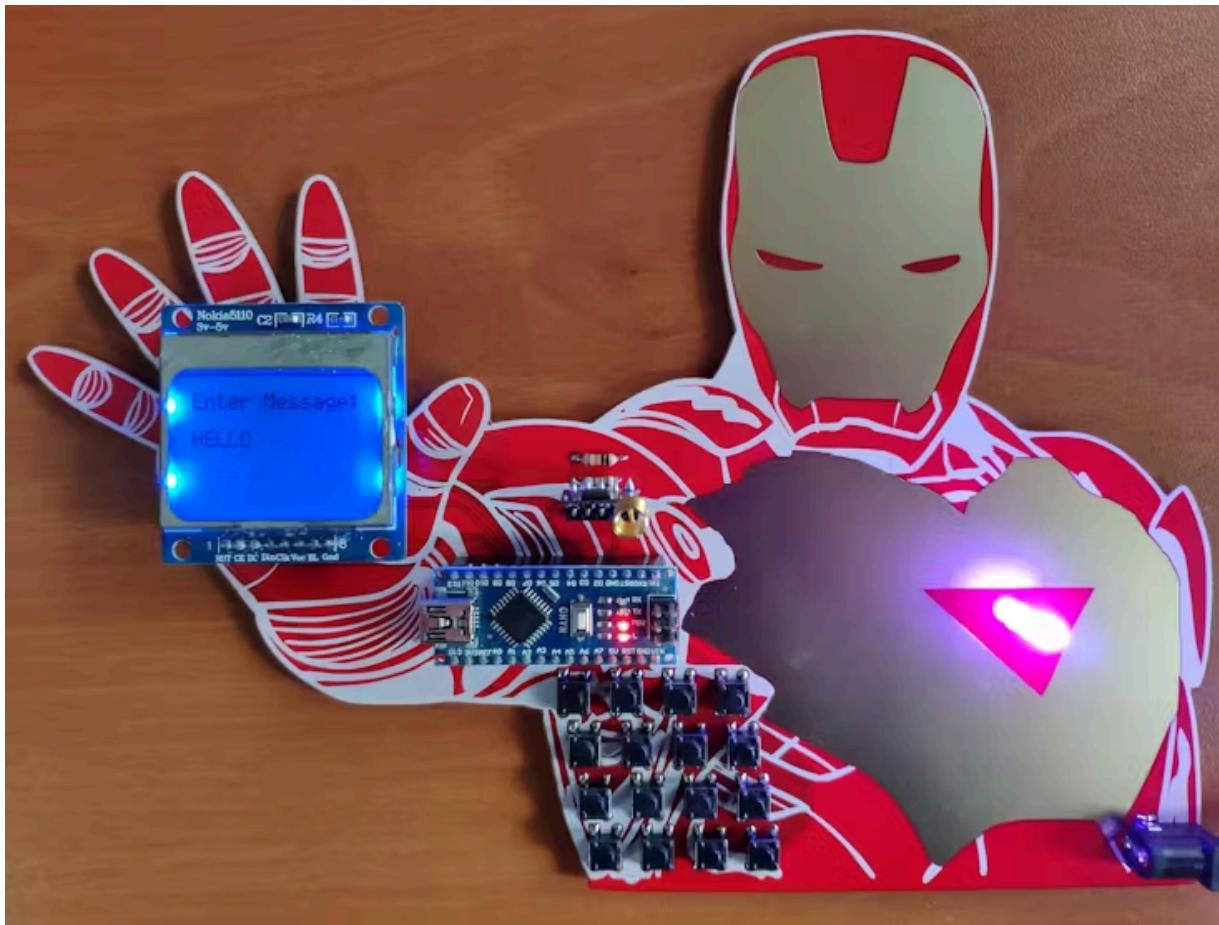


**Thanks for visiting Hackster!**

We would love to get some feedback from you.



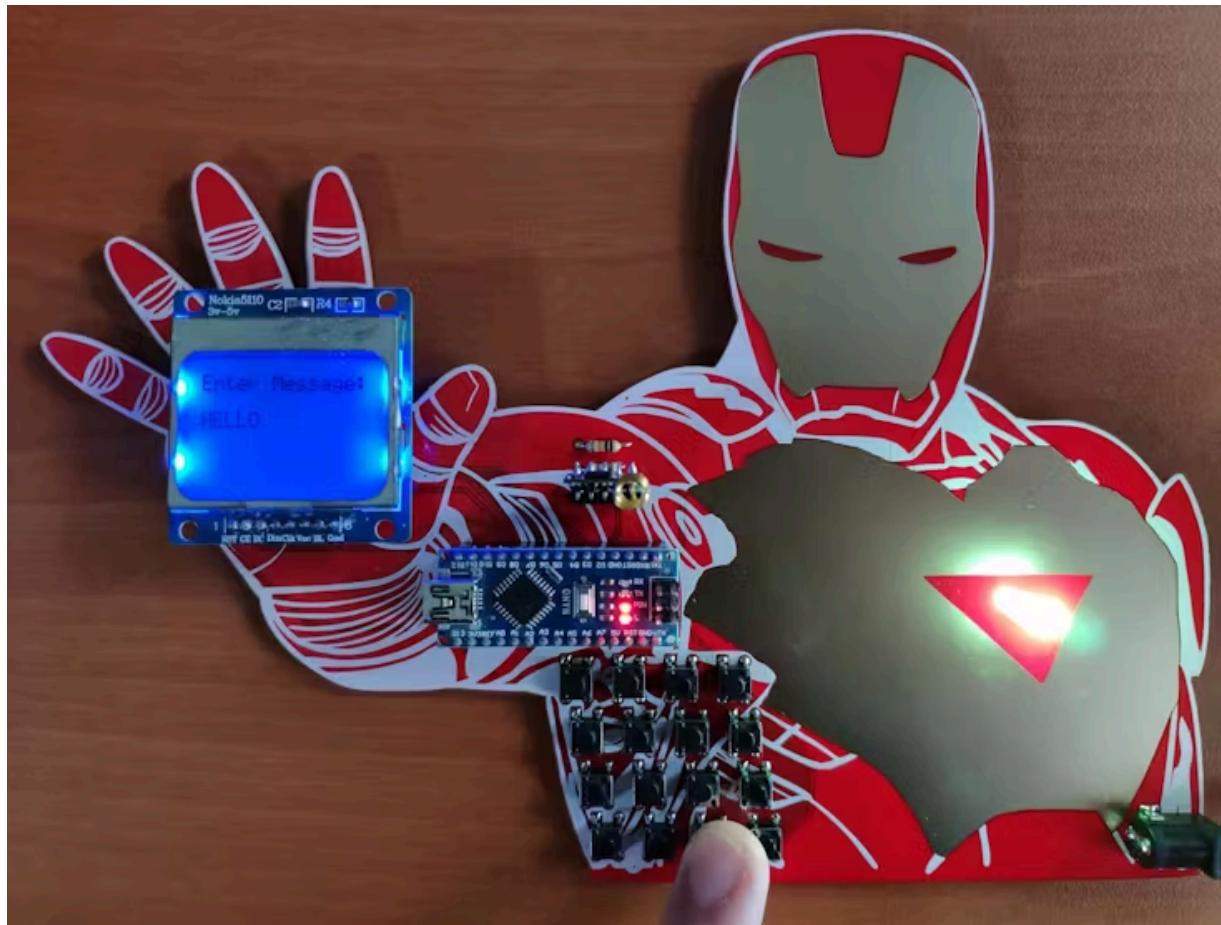
📞 After entering the message, the device lets the user scroll the entered message by pressing the '#' keypad button.



📞 The device transmits the recently entered message to the receiver Iron Man-themed walkie-talkie (two-way radio) if the '\*' keypad button is pressed.

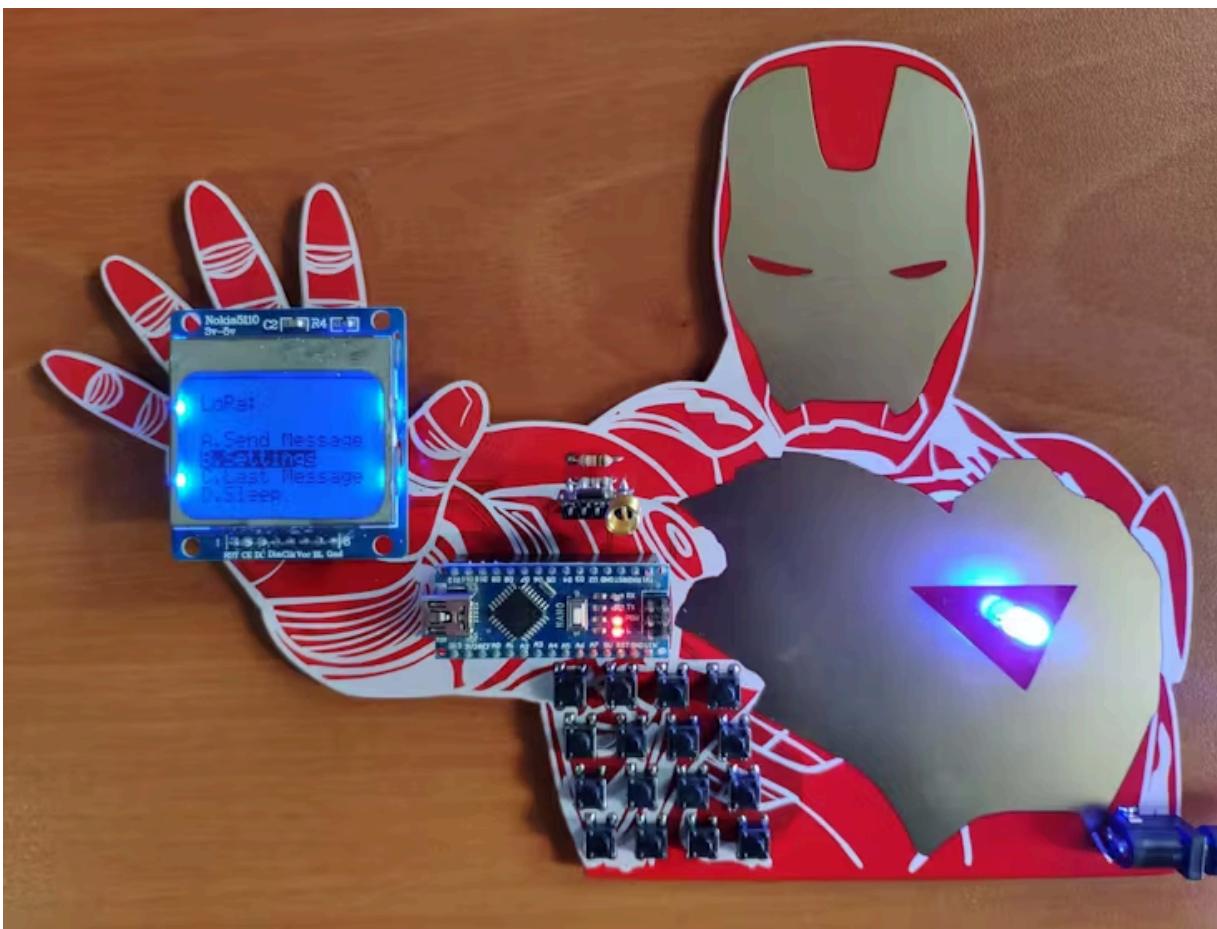
**Thanks for visiting Hackster!**  
We would love to get some feedback from you.

📞 If the given message is transmitted successfully to the receiver module, then the device makes the RGB LED blink yellow.



📌 **B. Settings**

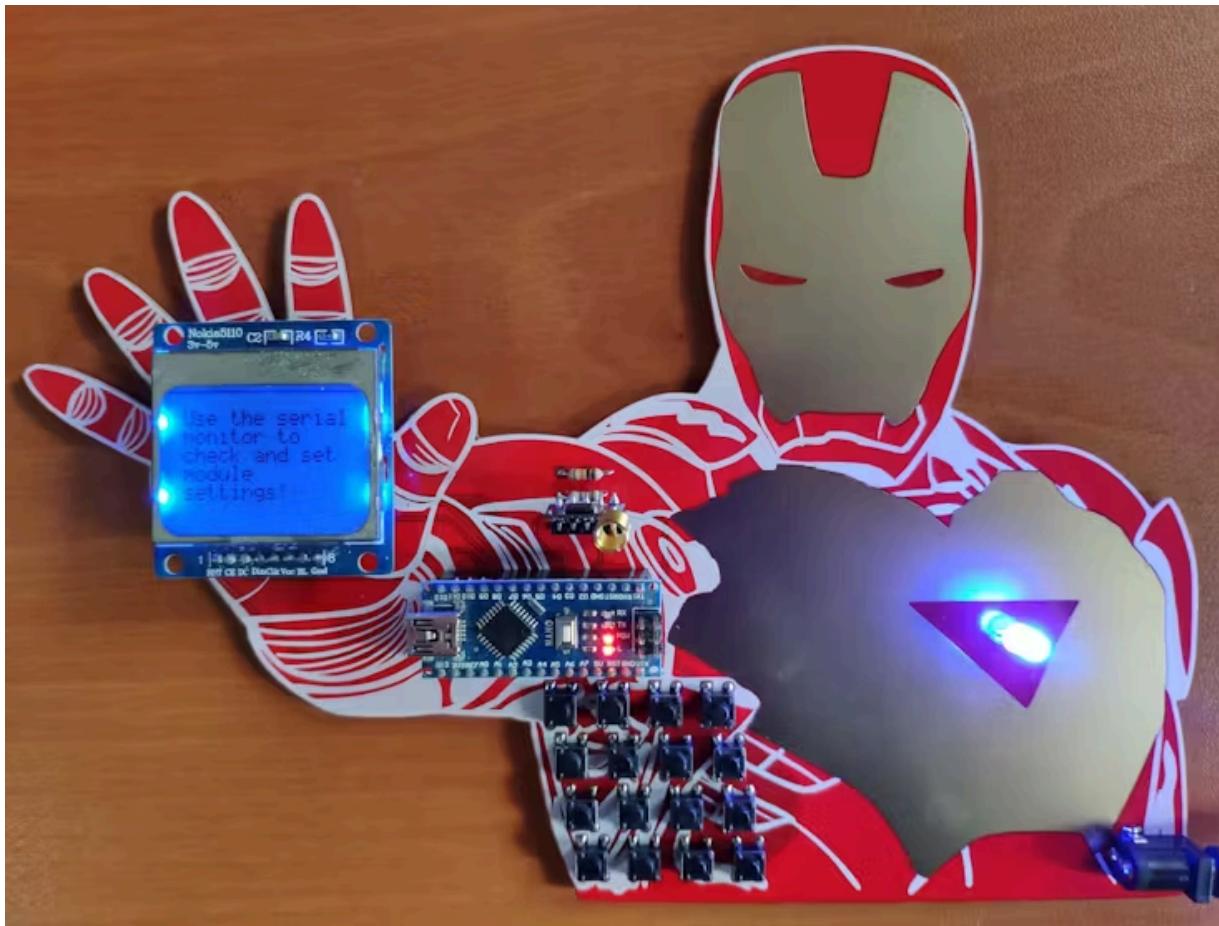
**Thanks for visiting Hackster!**  
We would love to get some feedback from you.



📞 The device allows the user to check and modify the RYLR998 LoRa module settings with AT commands on the serial monitor.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.



```
COM4
set
+OK
+OK
+OK
+OK
+OK
```

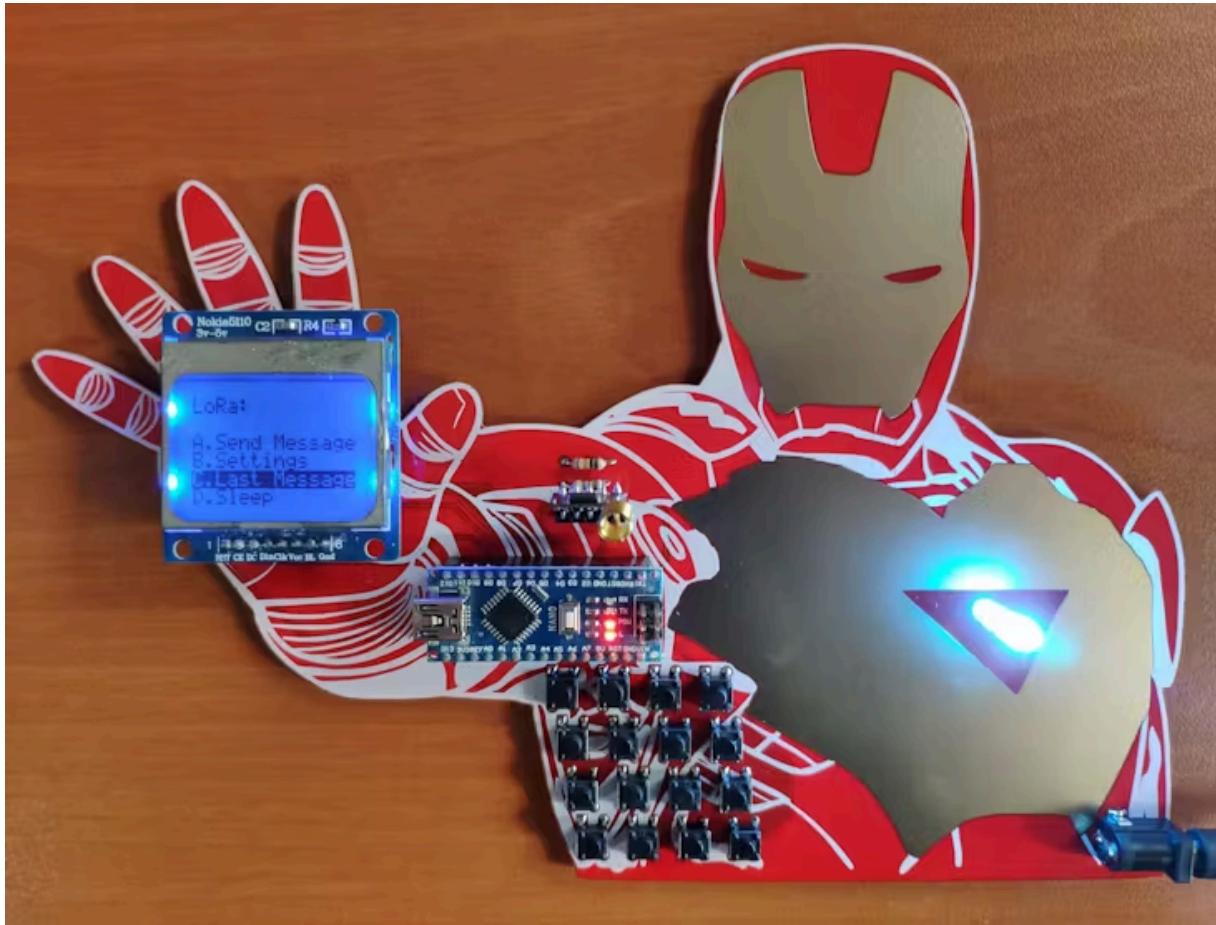
**Thanks for visiting Hackster!**

We would love to get some feedback from you.

```
COM4
[check]
+OK
+OK
+OK
+OK
+OK

+OK
+ADDRESS<108
+NETWORKID=10
+BAND|47000000
+PARAMETER=1
```

### 📌 C. Last Message

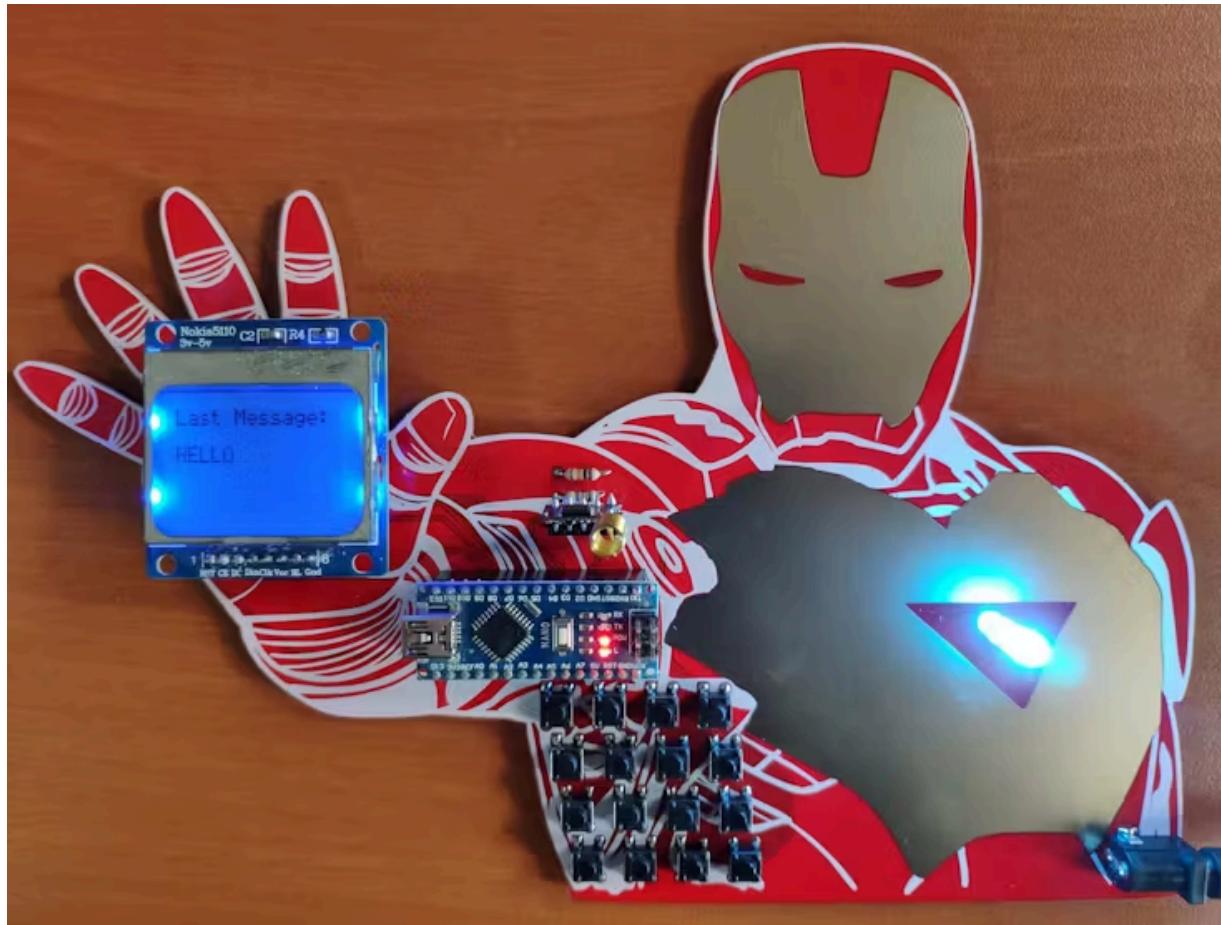


📱📞 The device displays the latest message transmitted to the receiver module if the '\*' keypad button is pressed.

**Thanks for visiting Hackster!**

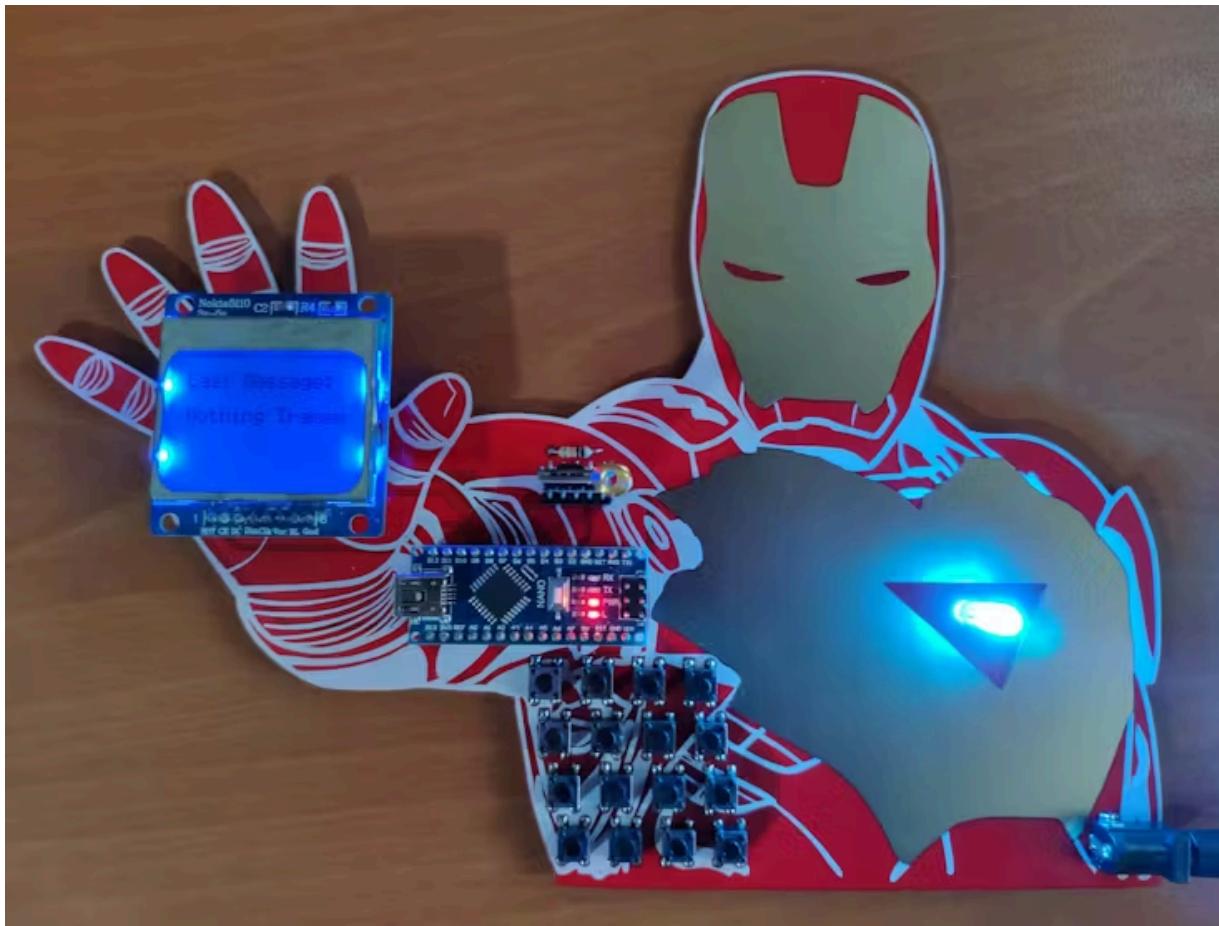
We would love to get some feedback from you.

📱 Also, the device lets the user scroll the latest transmitted message by pressing the '#' keypad button.



📱 If there is no previously transmitted message saved in the LoRa module's memory, then the device shows *Nothing Transmitted Yet!* as the latest message.

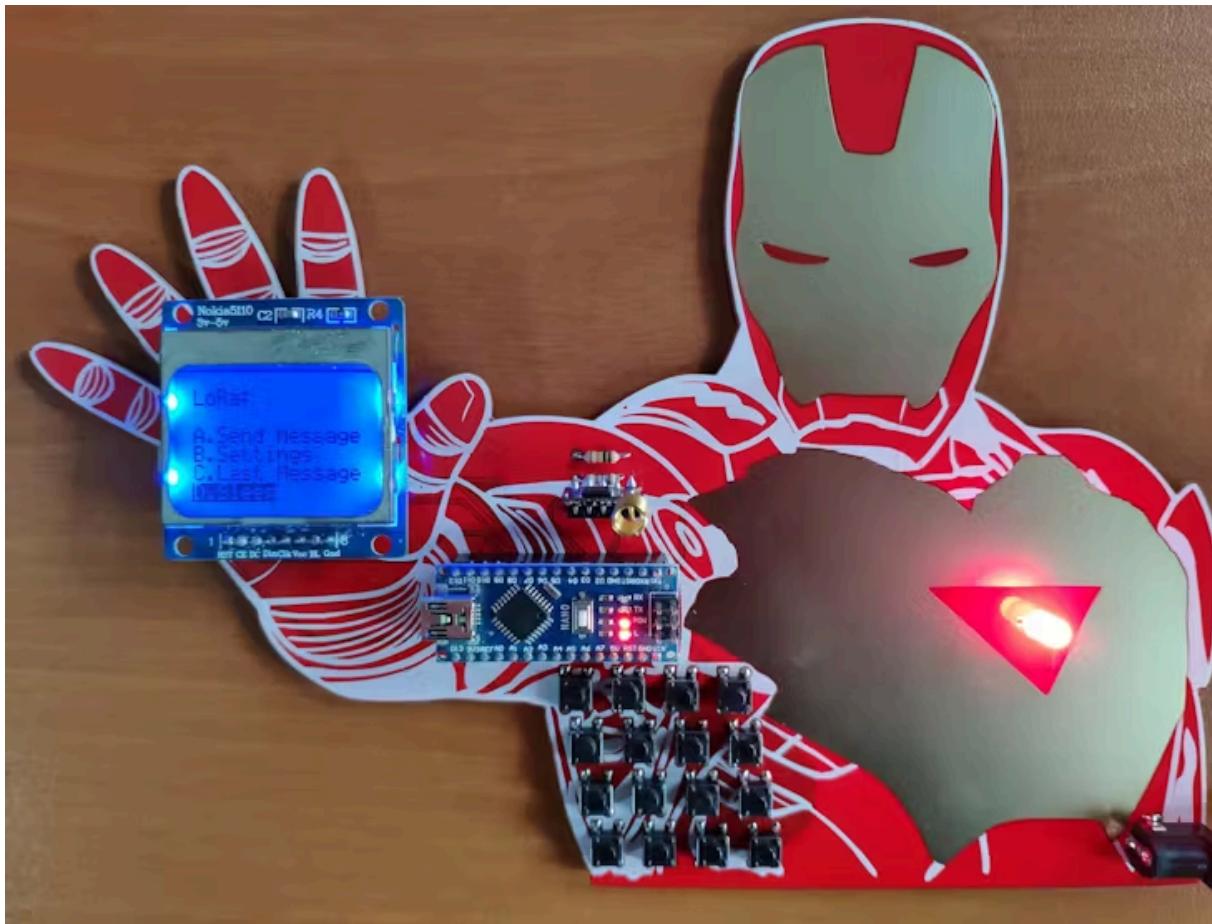
**Thanks for visiting Hackster!**  
We would love to get some feedback from you.



📌 D. Sleep

**Thanks for visiting Hackster!**

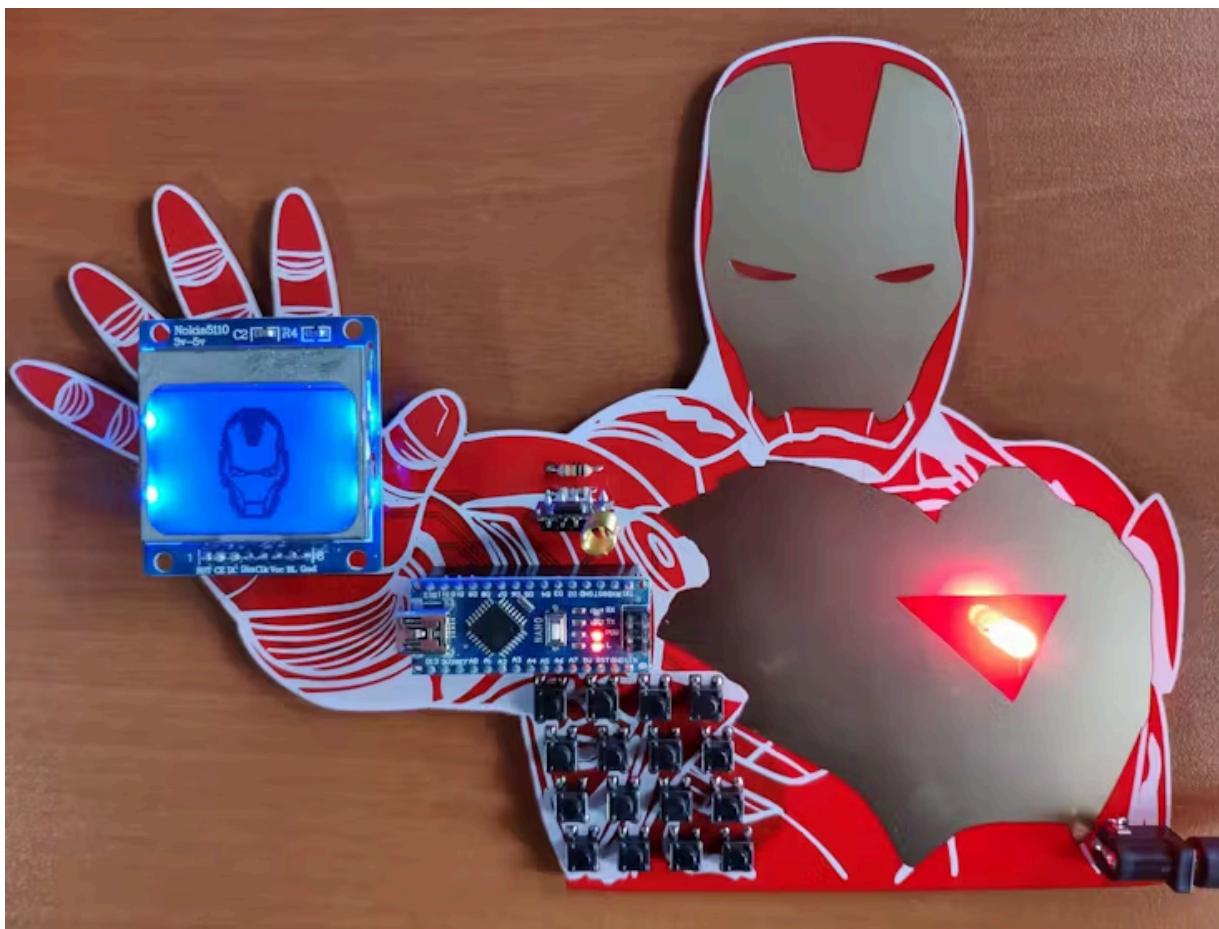
We would love to get some feedback from you.



📞 The device shows the Iron Man logo (monochrome image) as the screensaver while sleeping.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.



### 📌 Message Received

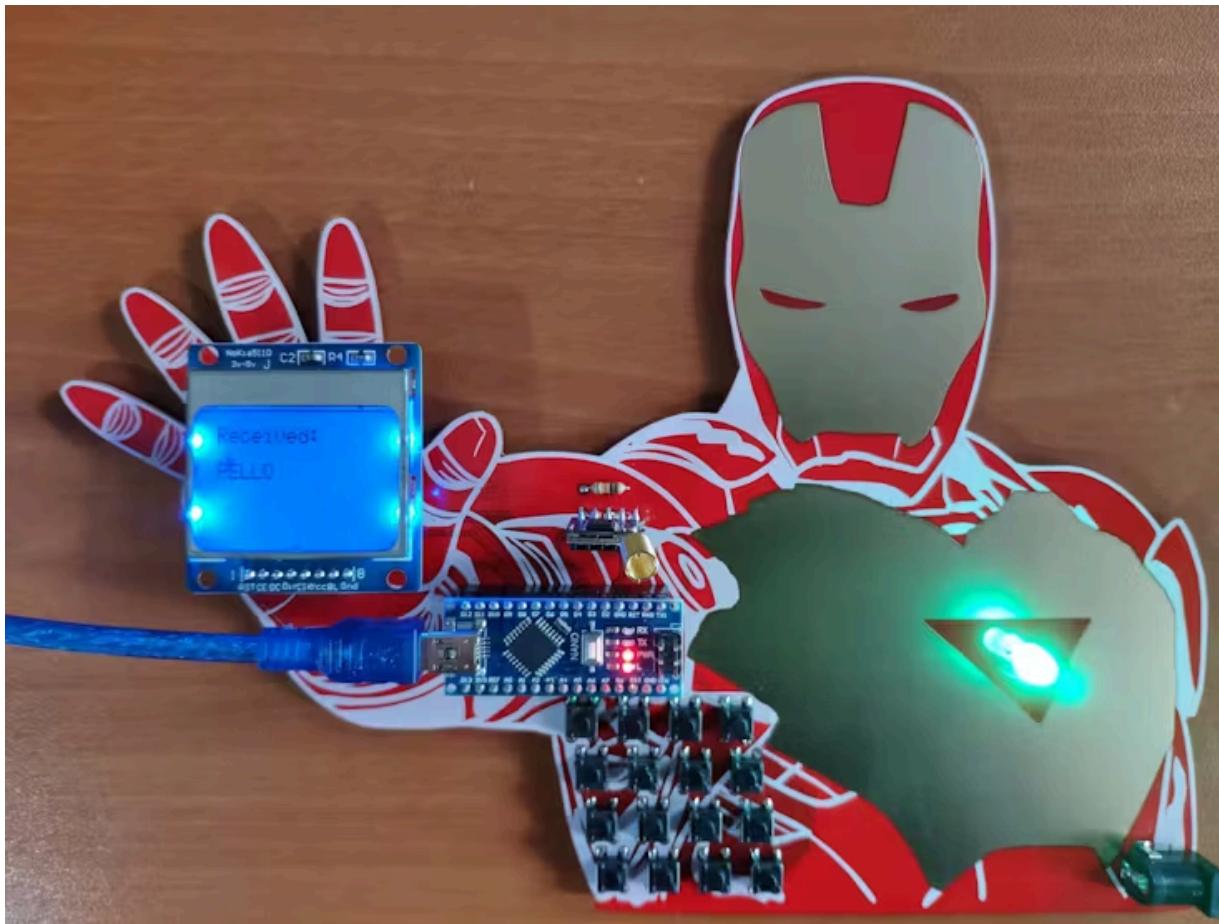
📞 The device shows the recently received message from the transmitter module. Then, it makes the RGB LED blink green.

📞 Also, the device lets the user scroll the received message by pressing the '#' keypad button.

📞 Until the '-' keypad button is pressed, the device shows the received message and keeps the RGB LED green.

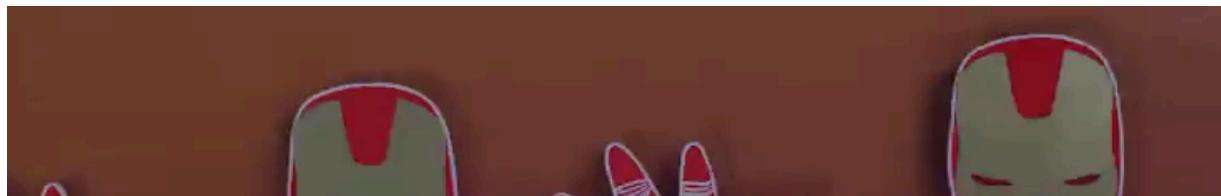
**Thanks for visiting Hackster!**

We would love to get some feedback from you.



**Thanks for visiting Hackster!**

We would love to get some feedback from you.



## Videos and Conclusion

Iron Man Walkie-Talkie (Two-Way Radio) for Texting w/ LoRa



After completing all steps above, I experimented with my Iron Man-themed walkie-talkies (two-way radios) to transmit and receive text messages. As far as my experiments go, I did not encounter any problems :)

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

---

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

# Custom parts and enclosures

G  
e  
r  
b  
e  
r  
F  
il  
e  
s

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396440/gerber\\_IF1aDHMIQI.zip](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396440/gerber_IF1aDHMIQI.zip))

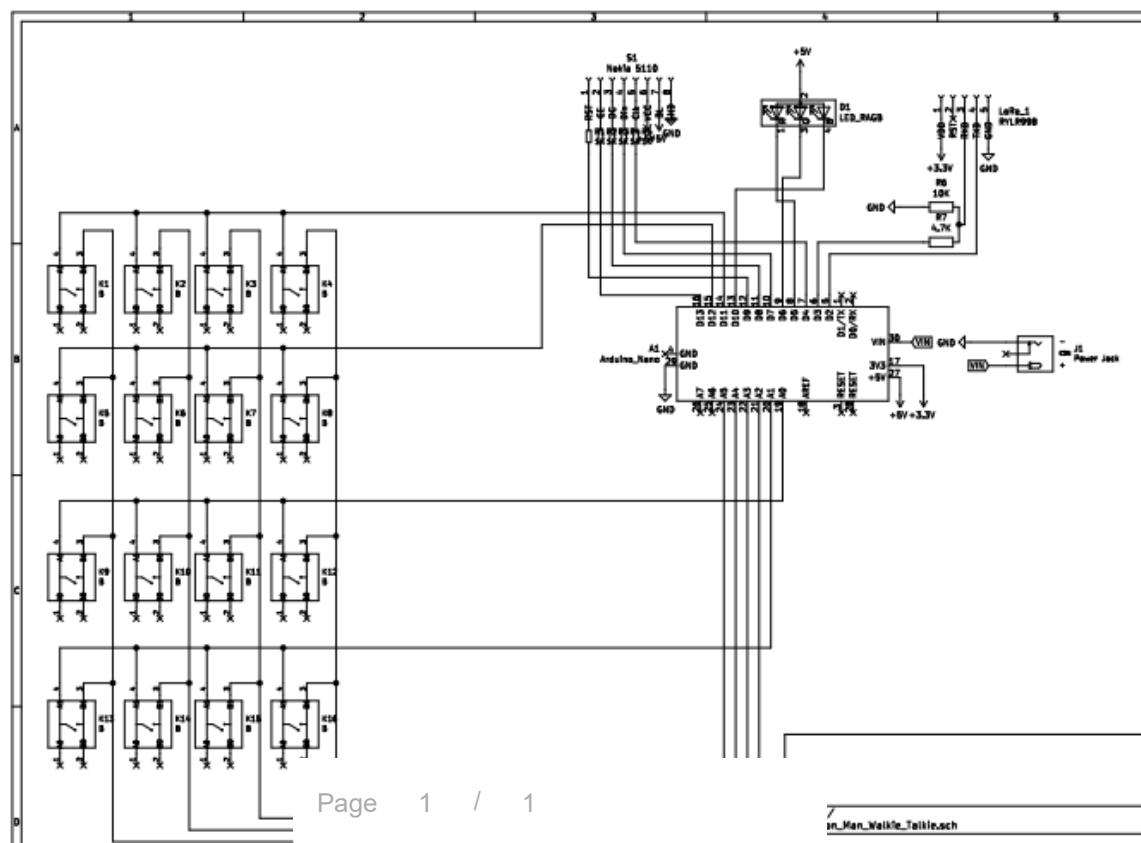
F  
a  
b  
r  
i  
c  
a  
t

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396441/fabrication\\_a1vmTT7fSP.pdf](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396441/fabrication_a1vmTT7fSP.pdf))

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

i  
o  
n  
F  
il  
e  
s



**Thanks for visiting Hackster!**

We would love to get some feedback from you.

ir  
o  
n  
—  
m  
a  
n.  
c

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396442/iron\\_man\\_NTjrJeH3Sb.c](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396442/iron_man_NTjrJeH3Sb.c))

---

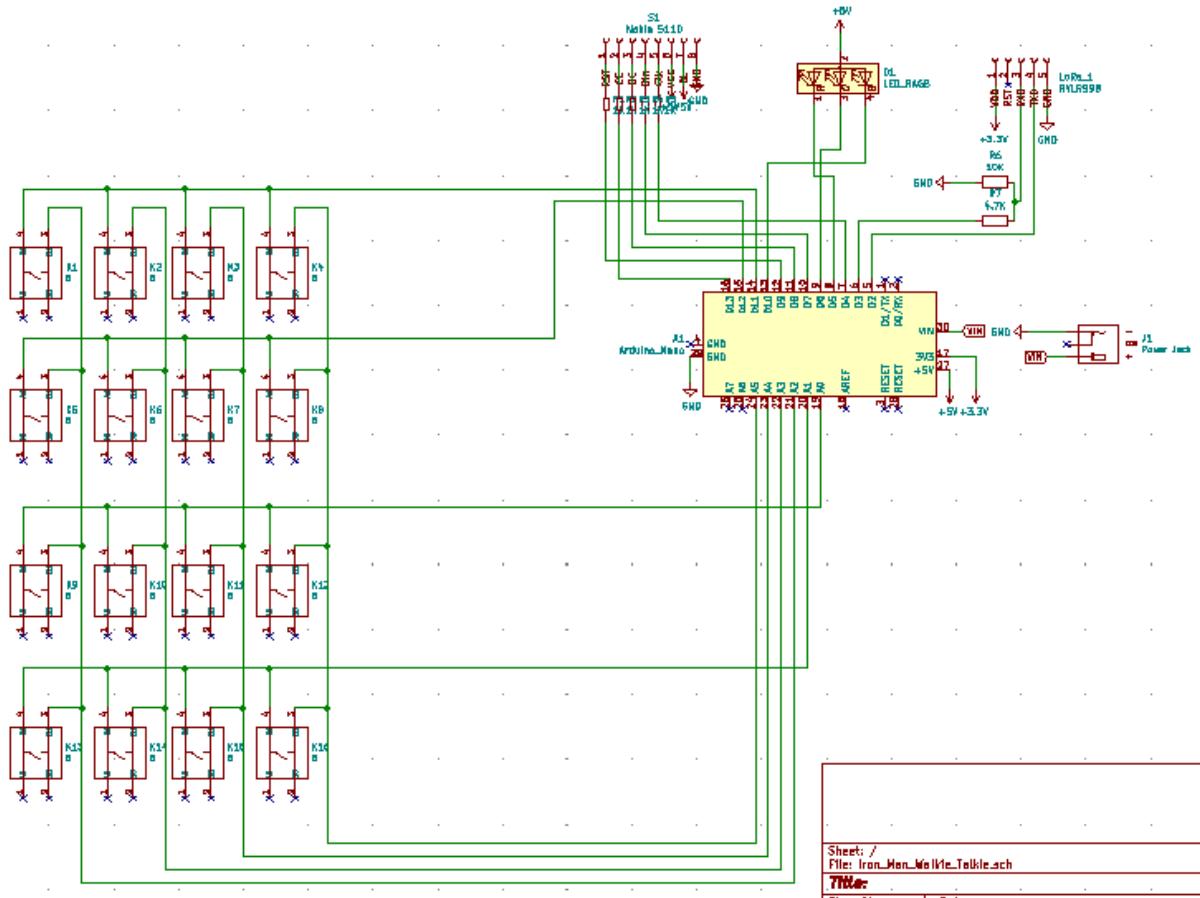
**Thanks for visiting Hackster!**

We would love to get some feedback from you.

# Schematics

P  
C  
B  
—  
1

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396434/pcb\\_1\\_xzwvjufpCO.PNG](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396434/pcb_1_xzwvjufpCO.PNG))

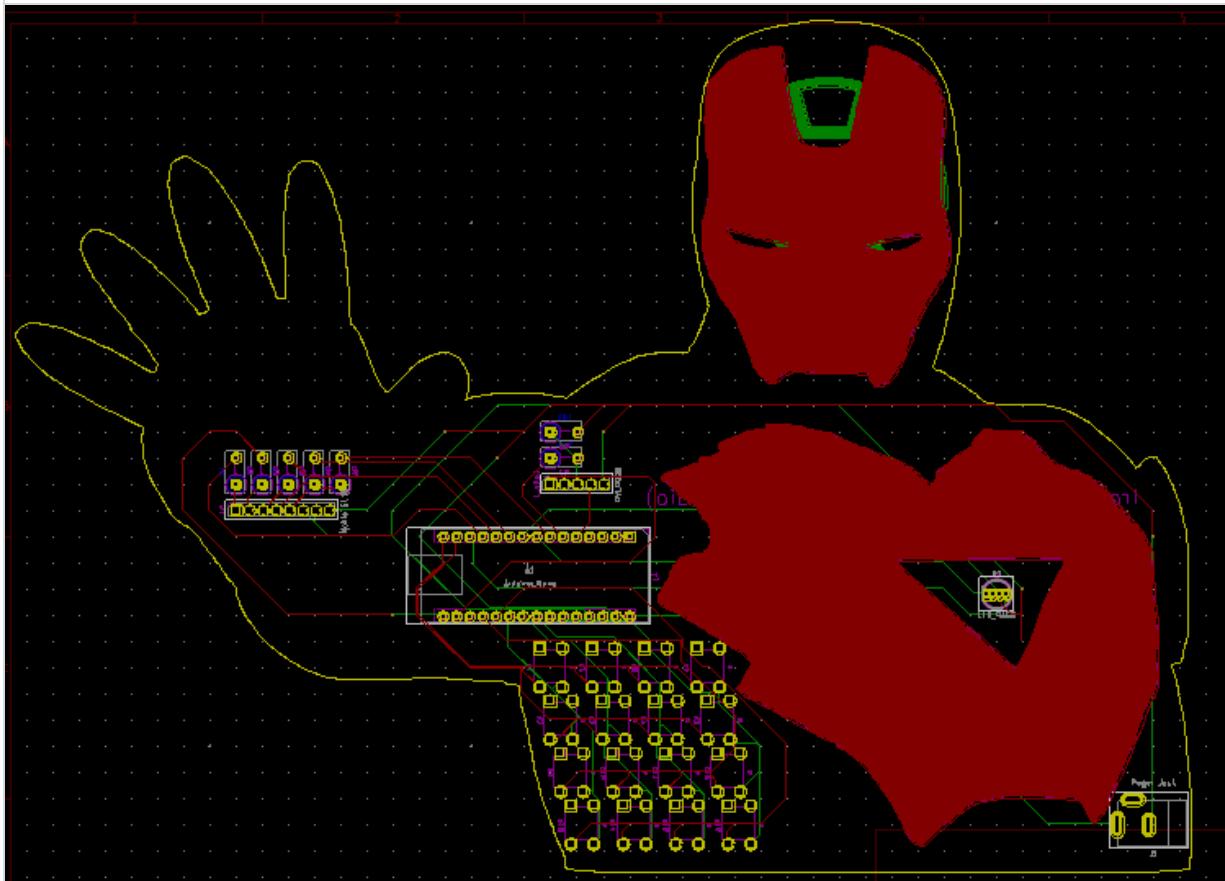


**Thanks for visiting Hackster!**

We would love to get some feedback from you.

P  
C  
B  
—  
2

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396435/pcb\\_2\\_DpjEXSh1hG.PNG](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396435/pcb_2_DpjEXSh1hG.PNG))

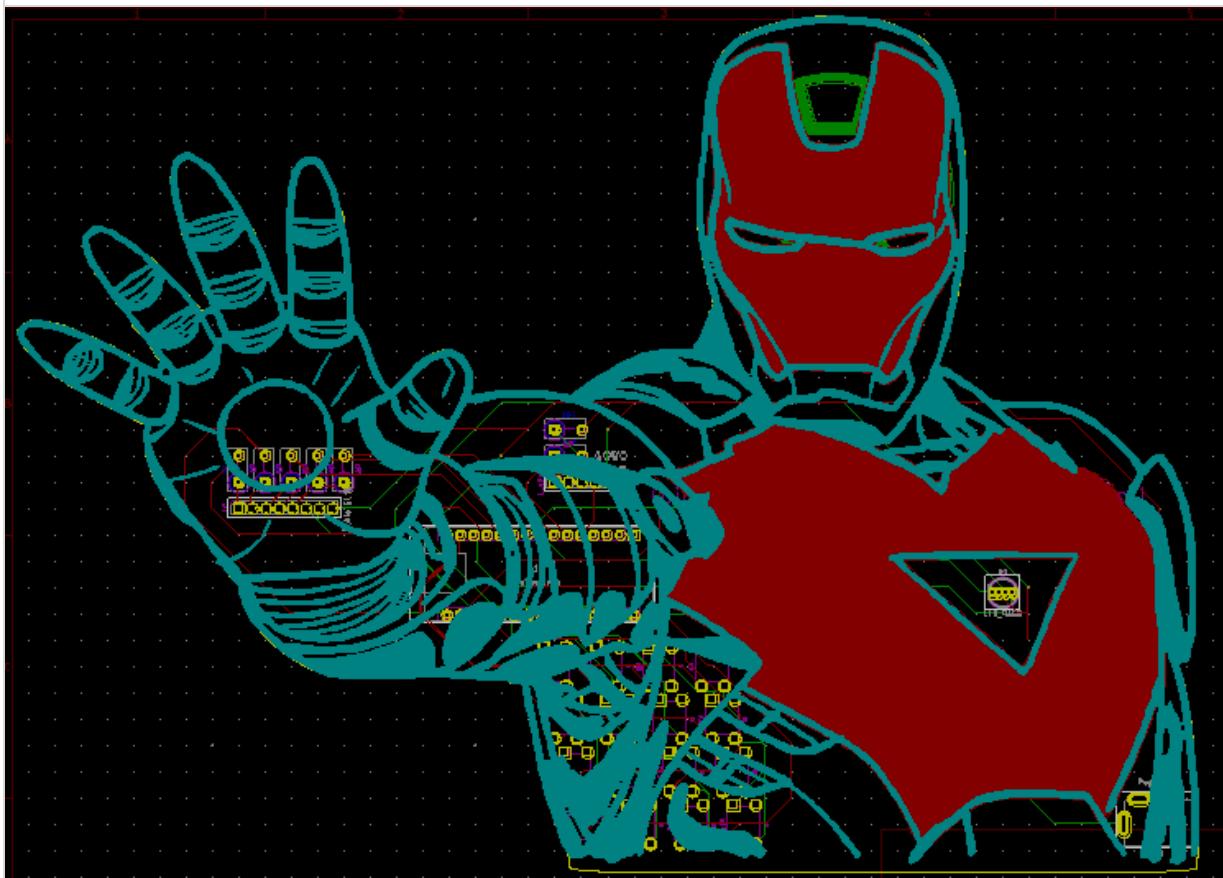


**Thanks for visiting Hackster!**

We would love to get some feedback from you.

P  
C  
B  
—  
3

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396436/pcb\\_3\\_wC2ZUKBidW.PNG](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396436/pcb_3_wC2ZUKBidW.PNG))

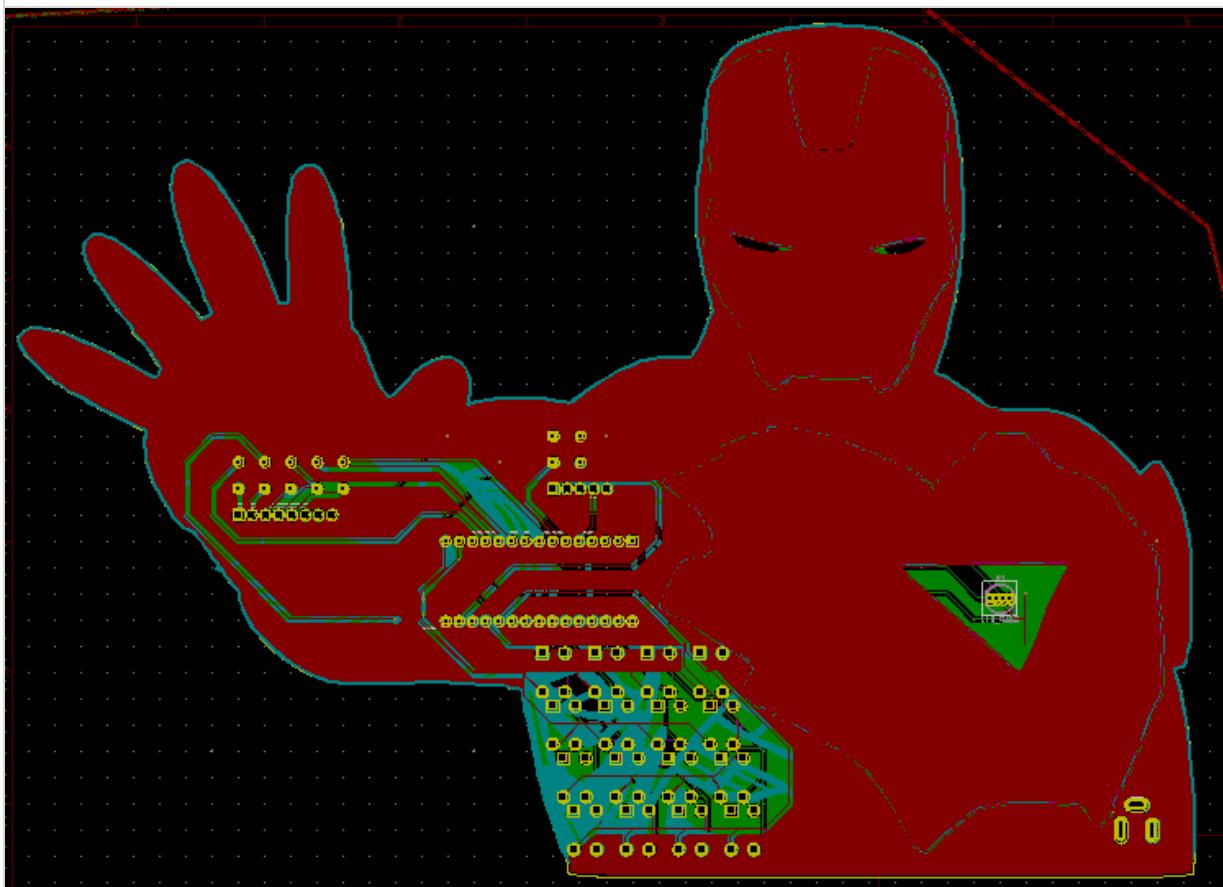


**Thanks for visiting Hackster!**

We would love to get some feedback from you.

P  
C  
B  
—  
4

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396437/pcb\\_4\\_kObQSrhNWz.PNG](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396437/pcb_4_kObQSrhNWz.PNG))

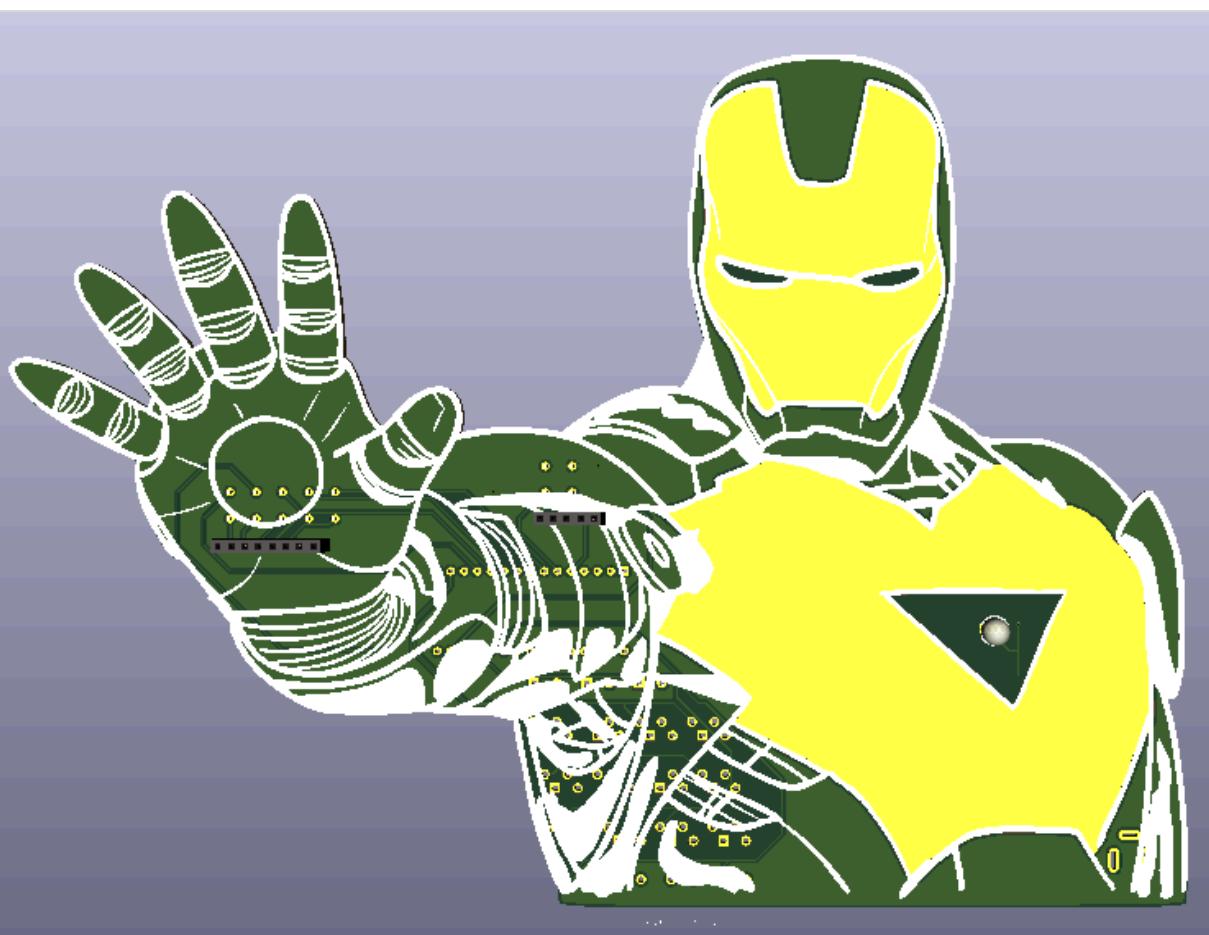


**Thanks for visiting Hackster!**

We would love to get some feedback from you.

P  
C  
B  
—  
5

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396438/pcb\\_5\\_bxAdhqF0ZR.PNG](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396438/pcb_5_bxAdhqF0ZR.PNG))

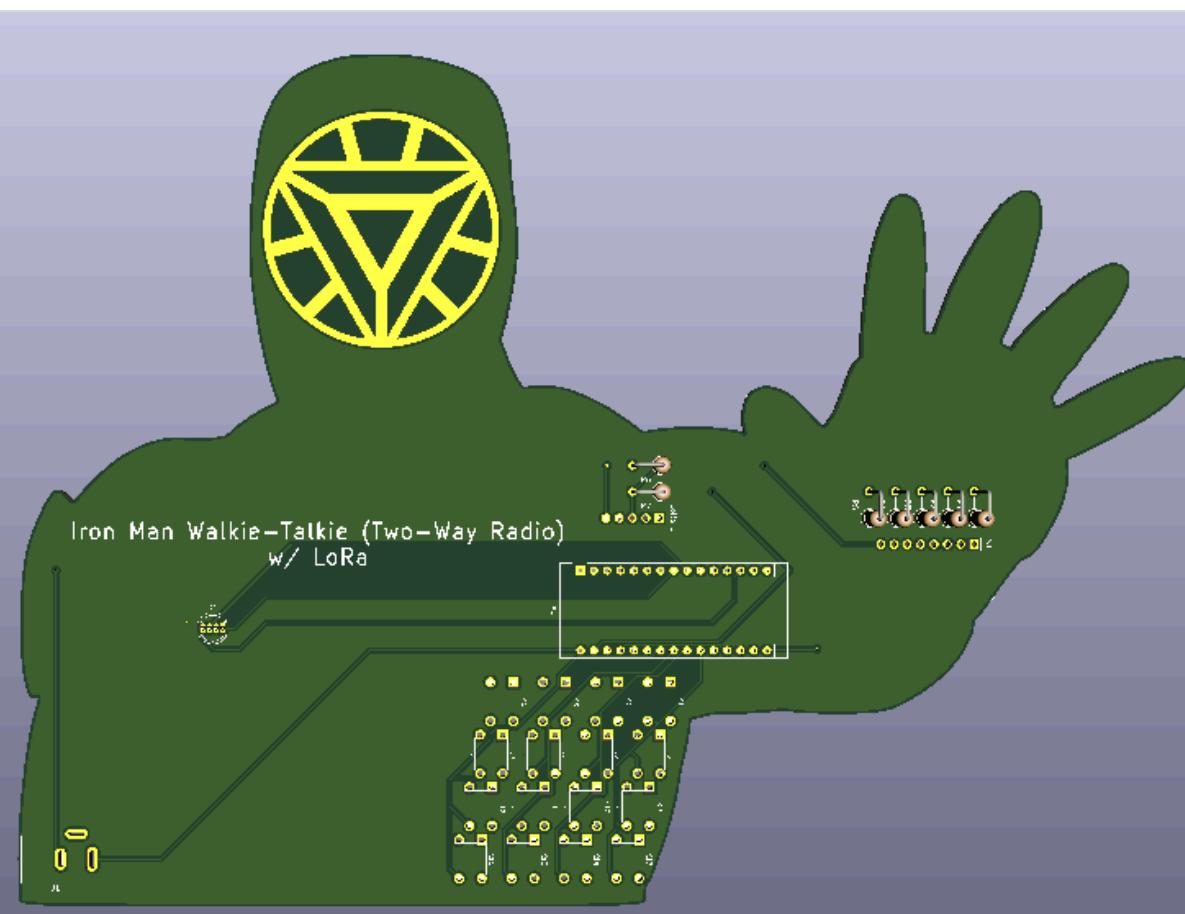


**Thanks for visiting Hackster!**

We would love to get some feedback from you.

P  
C  
B  
—  
6

([https://hacksterio.s3.amazonaws.com/uploads/attachments/1396439/pcb\\_6\\_Tjkqe630IY.PNG](https://hacksterio.s3.amazonaws.com/uploads/attachments/1396439/pcb_6_Tjkqe630IY.PNG))



**Thanks for visiting Hackster!**

We would love to get some feedback from you.

# Code

## Iron\_Man\_Walkie-Talkie\_w\_LoRa.ino

[\(/code\\_files/570169/download\)](#)

```
2      ///////////////////////////////////////////////////////////////////
3      // Iron Man Walkie-Talkie (Two-Way Radio) //
4      // w/ LoRa                                //
5      // -----                                //
6      // (Arduino Nano)                         //
7      // by Kutluhan Aktar                      //
8      //                                         //
9      ///////////////////////////////////////////////////////////////////
10
11 //
12 // Via RYLR998 LoRa module and Arduino Nano, create a two-way radio (transceiver)
13 //
14 // For more information:
15 // https://www.theamplituhedron.com/projects/Iron_Man_Walkie_Talkie_Two_Way_Radio
16 //
17 //
18 // Connections
19 // Arduino Nano :
20 //                               Nokia 5110 Screen
21 // D4 ----- SCK (Clk)
22 // D7 ----- MOSI (Din)
23 // D8 ----- DC
24 // D9 ----- RST
25 // D12 ----- CS (CE)
```

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

# Credits



## Kutluhan Aktar (/kutluhan-aktar)

81 projects • 303 followers

(/kutluhan-aktar) AI & Full-Stack Developer | @EdgelImpulse | @Particle | Maker | Independent Researcher

[Follow](#)

Contact (/users/sign\_up?  
redirect\_to=%2Fmessages%2Fnew%3Frecipient\_id%3D223436&source=user\_co

# Comments

Please log in or sign up to comment.

**Thanks for visiting Hackster!**

We would love to get some feedback from you.

**About Us**

[Hackster Overview \(/about\)](#)  
[Hackster for Business \(/business\)](#)  
[Help Articles \(https://help.hackster.io\)](#)  
[Brand Resources \(/branding\)](#)  
[Sitemap \(/sitemap.xml.html\)](#)

**Legal Thingies**

[Terms of Service \(/terms\)](#)  
[Code of Conduct \(/conduct\)](#)  
[Privacy Policy \(/privacy\)](#)  
[Privacy Policy for California](#)  
[Residents \(/privacy/ccpa\)](#)  
[Cookie Policy \(/cookies\)](#)

**Find Us On Social**

 [Facebook](#)  
(https://www.facebook.com/hacksterio)  
 [Instagram](#)  
(https://www.instagram.com/hacksterio)  
 [LinkedIn](#)  
(https://www.linkedin.com/company/hacksterio)  
 [X](#)  
(https://www.twitter.com/hacksterio)  
 [YouTube](#)  
(https://www.youtube.com/hacksterio)

**Visit Our Avnet Family**

[Avnet \(https://www.avnet.com\)](#)  
[Farnell \(https://www.farnell.com\)](#)  
[Element14 \(https://www.element14.com\)](#)  
[Newark \(https://www.newark.com\)](#)

Hackster.io, an Avnet Community © 2024

**Thanks for visiting Hackster!**

We would love to get some feedback from you.