<https://idyl.io/arduino/how-to/syn115-syn480r-rf-wireless-transmitter-receiver/>

TZT-Módulo de placa de receptor y transmisor inalámbrico SYN115 SYN480R ASK/OOK, PCB para arduino, 433MHZ, 1 Juego

* Data rate up to 10kbits ASK protocol
* Extreme low power (1.8 to 3.6V)
* 300 to 450Mhz operating range
* ~12.3 mA of current when transmitting data
* ~3 mA standby current

GOOD

- Low power RF!

- Very small

BAD

- Unreliable, you have to check

- Slow

- Unusable without an antenna ???

Use bluetooth for sending

* Lot of data
* Faster data rate



Cuando presione el boton enviara senal binaria para encender o apagar led

Encender 0101 0101 Hex 0x55

Apagar 1001 0110 Hex 0x96

|  |  |
| --- | --- |
| Transmisor | Receptor |
| IMPORTANTE, COMO CHIP FUNCIONA A 3.3V DEBE REDUCIR EL VOLTAJE, O UTILIZA UN ESP32 QUE SUS PINES GENERAN 3.3V   * 51.2kΩ resistor (top) x1 & 100KΩ resistor (botton)   Como transmite, reduce el voltaje de 5v que sale de PIN 8 para la senal de entrada | * 330Ω to 1kΩ resistor x1 (used to limit current to the LED) |

Usa resistencia para divider to prevent the 5v data line from the arduino from damaging the SYN115 which is running at 3.3v.

I am 99% sure that I have used those transmitters at 5v without issue.

Transmisor

//Uses idyl.io Button library. Feel free to replace it by raw reading or your own version

#include <Button.h>

#include <RH\_ASK.h>

uint8\_t ledStatus = false;

const char\* ON\_MESSAGE = new char[1]{0x55}; //binary 0101 0101

const char\* OFF\_MESSAGE = new char[1]{0x96}; //binary 1001 0110

//setup the driver as such:

// speed: 2400 bits per second

// rxPin - receiving pin: 11 --not used for this transmitting script

// txPin - transmitting pin: 12

// pttPin - transmitting enable pin: 10 --not used for this transmitting script.

// RadioHead can enable the transmitter only when transmitting data

// but this is not supported on the SYN115 breakout board

// pttInverted: true - LOW signal will enable the tranmitter if set to true.

// Since we don't control the transmitter

// we want this value to be true

RH\_ASK driver(2400, 11, 12, 10, true);

Button button(8, INPUT\_PULLUP, true);

void setup() {

//initialize wireless driver

driver.init();

}

void loop() {

button.poll();

//send one byte. To send a whole message you can use strlen(MESSAGE)

if(button.rising()){

ledStatus = !ledStatus; //turn LED on or off

if(ledStatus){

driver.send((uint8\_t \*)ON\_MESSAGE, 1);

}

else{

driver.send((uint8\_t \*)OFF\_MESSAGE, 1); //send one byte

}

driver.waitPacketSent();

}

}

There are two interesting features to note in the snippet above:

* We are transmitting a single byte; hence the “1” in the driver.send call.
* The byte for ON and OFF are respectively 0101 0101 and 1001 0110.
* byte such as 0000 0000 is not so easy to transmit over the air.

Codigo sin boton

#include <RH\_ASK.h>

uint8\_t ledStatus = false;

const char\* ON\_MESSAGE = new char[1]{0x55}; //binary 0101 0101

const char\* OFF\_MESSAGE = new char[1]{0x96}; //binary 1001 0110

RH\_ASK driver(2400, 11, 12, 10, true);

void setup() {

driver.init();

}

void loop() {

ledStatus = !ledStatus; //turn LED on or off

if(ledStatus){

driver.send((uint8\_t \*)ON\_MESSAGE, 1);

}

else{

driver.send((uint8\_t \*)OFF\_MESSAGE, 1); //send one byte

}

driver.waitPacketSent();

delay(2000);

}

Receptor, recibe senal binaria, enciende y apaga LED

#include <RH\_ASK.h>

RH\_ASK driver(2400, 11, 12, 10, true);

void setup()

{

pinMode(8, OUTPUT);

driver.init();

}

void loop()

{

uint8\_t buf[**RH\_ASK\_MAX\_MESSAGE\_LEN**];

uint8\_t buflen = sizeof(buf);

if (driver.recv(buf, &buflen)) // Non-blocking

{

if(buflen >= 1){

if(buf[0] == 0x96){

digitalWrite(8, LOW);

}

else if(buf[0] == 0x55){

digitalWrite(8, HIGH);

}

}

}

}