ACTIVIDAD N°7

MODULO 3: ARQUTECTURA EN REDES IOT DE BAJO CONSUMO

CONTENIDO

DIRECCIONES

REPOSITORIO: HTTPS://GITHUB.COM/ISPC-OPALO/ARQUITECTURA-Y-CONECTIVIDAD

REQUISITOS: ../A REQUISITOS\ACTIVIDAD N7.PDF

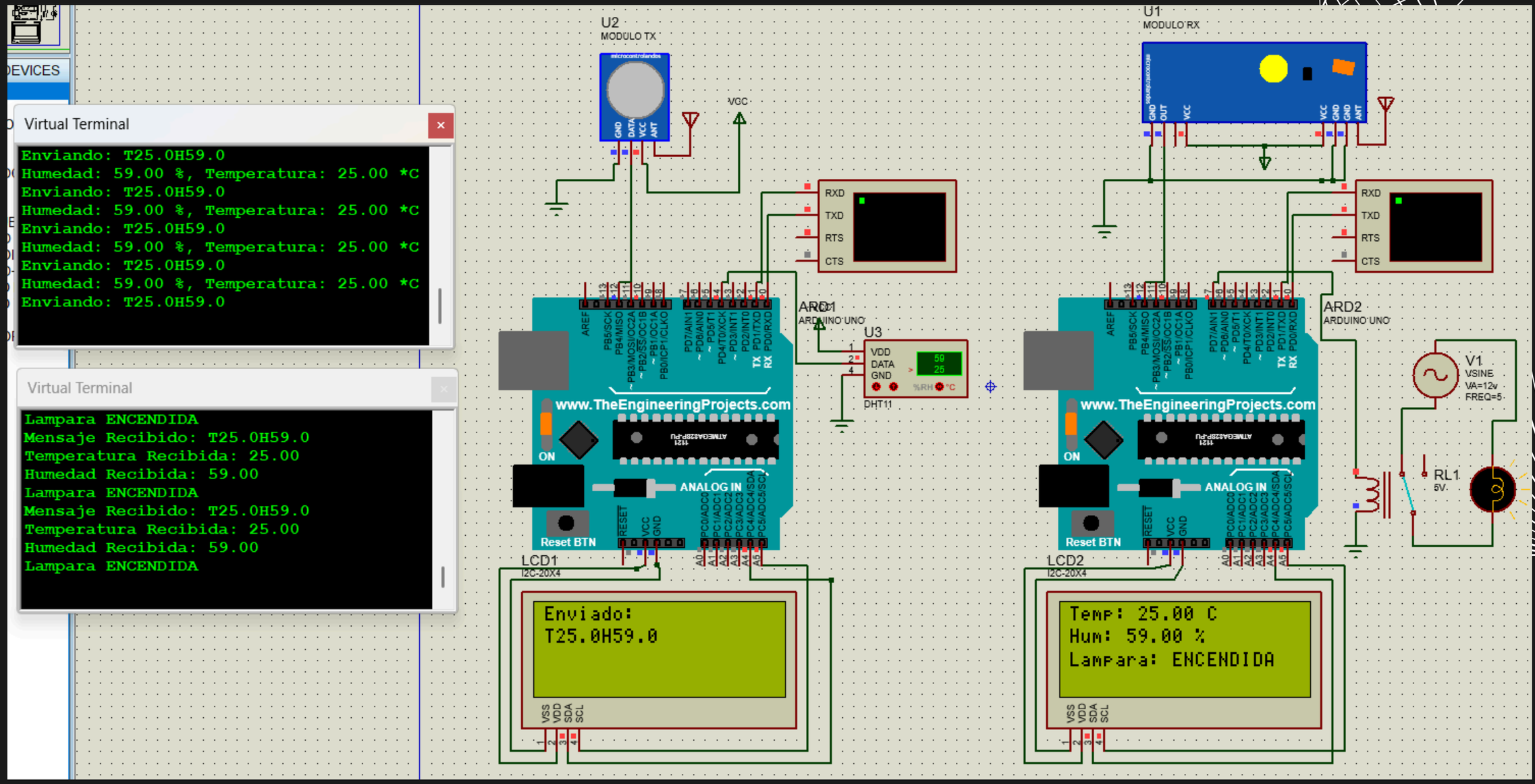
DATASHEETS: ../B INVESTIGACION\DATASHEETS

PROTOTIPO:

- **CODIGO TX:** ../C PROTOTIPO\ACTIVIDAD N°7\TP7 - RF\01-TRANSMISORRF-AYC
- **CODIGO RX:** ../C PROTOTIPO\ACTIVIDAD N°7\TP7 - RF\02-RECEPTORRF-AYC
- **MODELO PROTEUS:** ../C PROTOTIPO\ACTIVIDAD N°7\TP7 - RF\TP7-AYC2025-RF.PDSPRJ

PRESENTACION: ../D PRESENTACION\ACTIVIDAD N°7\ACTIVIDAD N°7.PPTX

VISTA FINAL

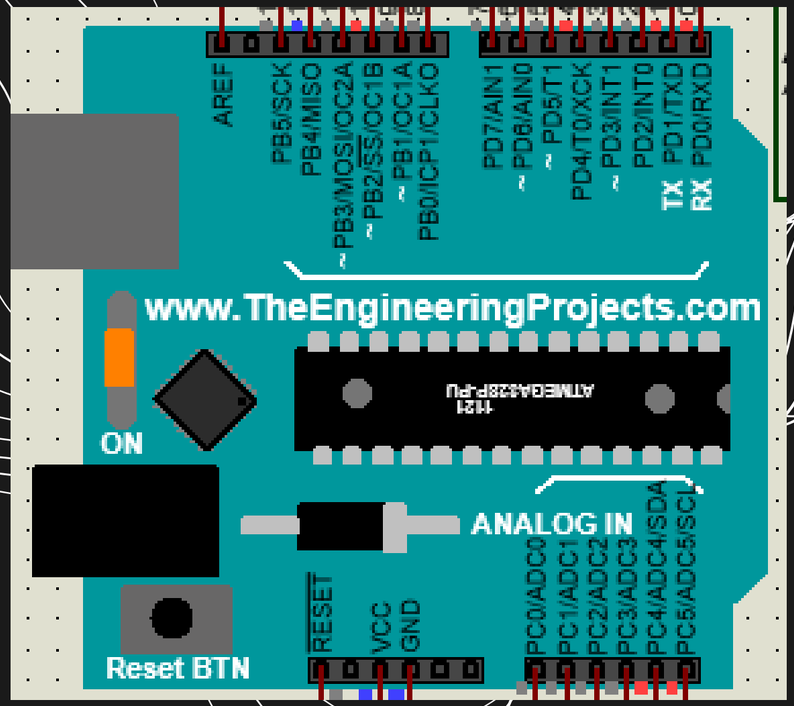

Virtual Terminal
Enviando: T25.0H59.0
Humedad: 59.00 %, Temperatura: 25.00 *C
Enviando: T25.0H59.0
Humedad: 59.00 %, Temperatura: 25.00 *C
Enviando: T25.0H59.0
Humedad: 59.00 %, Temperatura: 25.00 *C
Enviando: T25.0H59.0
Humedad: 59.00 %, Temperatura: 25.00 *C
Enviando: T25.0H59.0
Virtual Terminal
Lampara ENCENDIDA
Mensaje Recibido: T25.0H59.0
Temperatura Recibida: 25.00
Humedad Recibida: 59.00
Lampara ENCENDIDA
Mensaje Recibido: T25.0H59.0
Temperatura Recibida: 25.00
Humedad Recibida: 59.00
Lampara ENCENDIDA
U2
MODULO TX
U1
MODULO RX
VCC
RXD
TXD
RTS
CTS
ARD1
ARD2
ARDUINO UNO
U3
DHT11
www.TheEngineeringProjects.com
ON
ANALOG IN
Reset BTN
LCD1
I2C-20X4
Enviado:
T25.0H59.0
LCD2
Temp: 25.00 C
Hum: 59.00 %
Lampara: ENCENDIDA
V1
VSINE
VA=12v
FREQ=5
RL1
5V

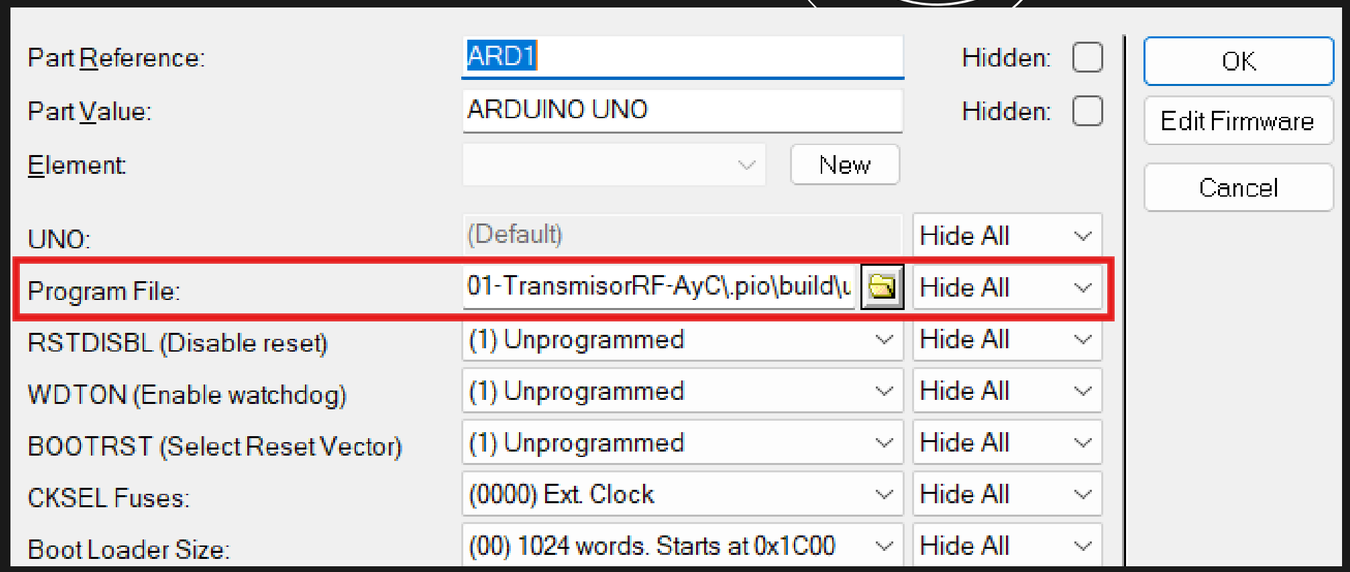
MICROCONTROLADOR

Se utilizo el arduino UNO, por que el modulo ya existe en la plataforma de proteus, y es facilmente configurable, partiendo el archivo complilado .hex realizado en otra plataforma

El archivo en formato hexadecimal que requiere el entrono de proteus lo optenemos mediante la complilacion realizada de los programas, tanto para el transmisor como el receptor, realizado en el entorno habitual de VsCode/PlatformIO

En la proxima diapositiva verificamos el codigo en cuestion.





CODIGO: TRANSMISOR RF

LIBRERIAS

```
#include <Wire.h>
#include <DHT.h>
#include <SPI.h>
#include <LiquidCrystal_I2C.h>
#include <VirtualWire.h>
```

CONFIGURACIONES DE PINES

```
#define DATA_PIN 12
#define DHTPIN 4
#define DHTTYPE DHT11
```

OBJETOS

```
LiquidCrystal_I2C lcd(0x38, 20, 4);
DHT dht(DHTPIN, DHTTYPE);
```

INICIALIZAMOS ELEMTOS

VIRTUAL WIRE

```
vw_set_ptt_inverted(true);
vw_set_tx_pin(DATA_PIN);
vw_setup(4000);
```

COMUNICACION SERIAL

```
Serial.begin(9600);
Serial.println("Transmisor RF con DHT");
```

DHT11

```
dht.begin();
```

LCD

```
lcd.begin(16, 2);
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("TX Listo (DHT)");
```

LOOP

LECTURA DHT11

```
float h = dht.readHumidity();
float t = dht.readTemperature();
```

MANEJO DE ERROR

```
if (isnan(h) || isnan(t)) {}
```

FORMATO DE MSJ Y ENVIO CON VW

```
String message = "T" + String(t, 1) + "H" + String(h, 1);
Serial.print("Enviando: ");
Serial.println(message);

vw_send((uint8_t *)message.c_str(), message.length());
vw_wait_tx();
```

ACTUALIZAMOS DISPLAY CON MSJ

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enviado:");
lcd.setCursor(0, 1);
lcd.print(message);
```

CODIGO: TRANSMISOR RF

LIBRERIAS

```
#include <VirtualWire.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
```

CONFIGURACIONES DE PINES

```
#define DATA_PIN 12
#define RELAY_PIN 7
```

OBJETOS

```
LiquidCrystal_I2C lcd(0x38, 20, 4);
```

INICIALIZAMOS ELEMTOS

VIRTUAL WIRE

```
vw_set_ptt_inverted(true);
vw_set_rx_pin(DATA_PIN);
vw_setup(4000);
vw_rx_start();
```

*HABILITANDO MODO ESCUCHA PERMANENTE

RELE

```
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, LOW);
```

LCD

```
Serial.begin(9600);
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Esperando mensaje");
```

LOOP

RECEPCION Y PROCESAMIENTO DEL MSJ

```
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;

if (vw_get_message(buf, &buflen)) {
  String receivedMessage = (char*)buf;
```

EXTRACCION Y FORMATO

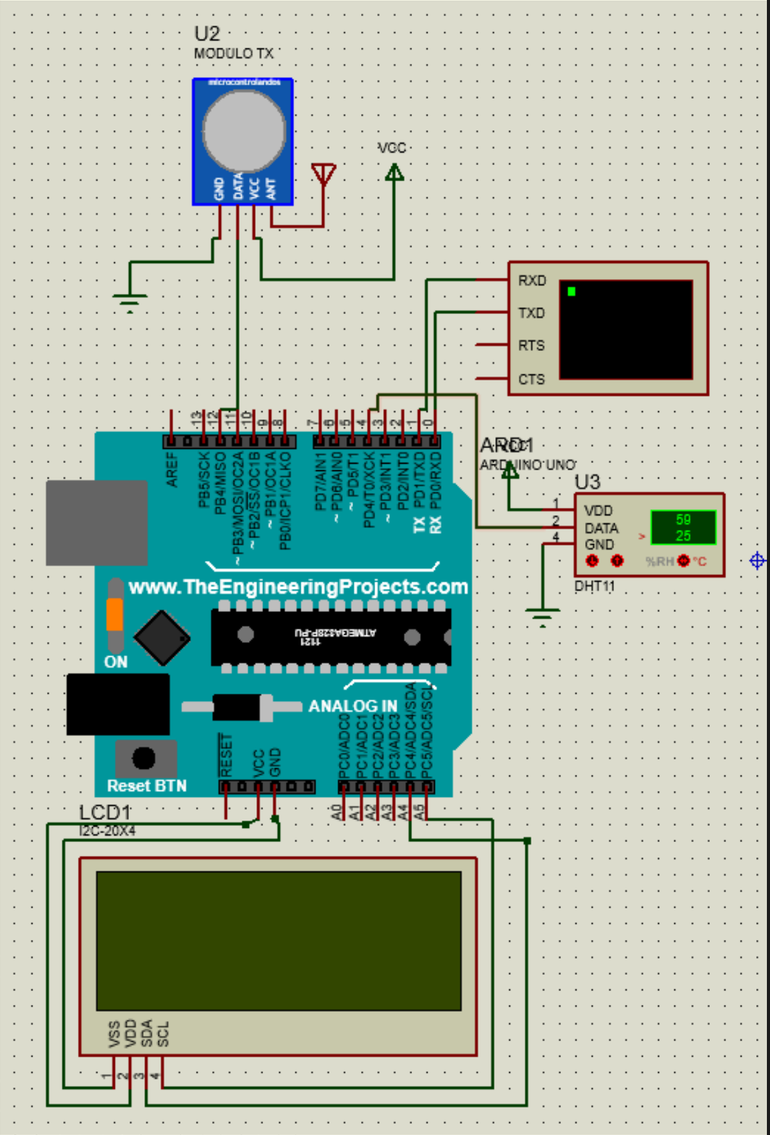
```
float temperature = -100.0;
float humidity = -1.0;
int tIndex = receivedMessage.indexOf('T');
int hIndex = receivedMessage.indexOf('H');

if (tIndex != -1 && hIndex != -1) {
  String tempStr = receivedMessage.substring(tIndex + 1, hIndex);
  temperature = tempStr.toFloat();
  String humStr = receivedMessage.substring(hIndex + 1);
  humidity = humStr.toFloat();
```

CONDICIONAL PARA MANEJO DE LA LAMPARA

```
if (temperature > 20.0 && humidity > 50.0) {
  digitalWrite(RELAY_PIN, HIGH); // Enciende la lámpara
  Serial.println("Lampara ENCENDIDA");
  lcd.setCursor(0, 2);
```

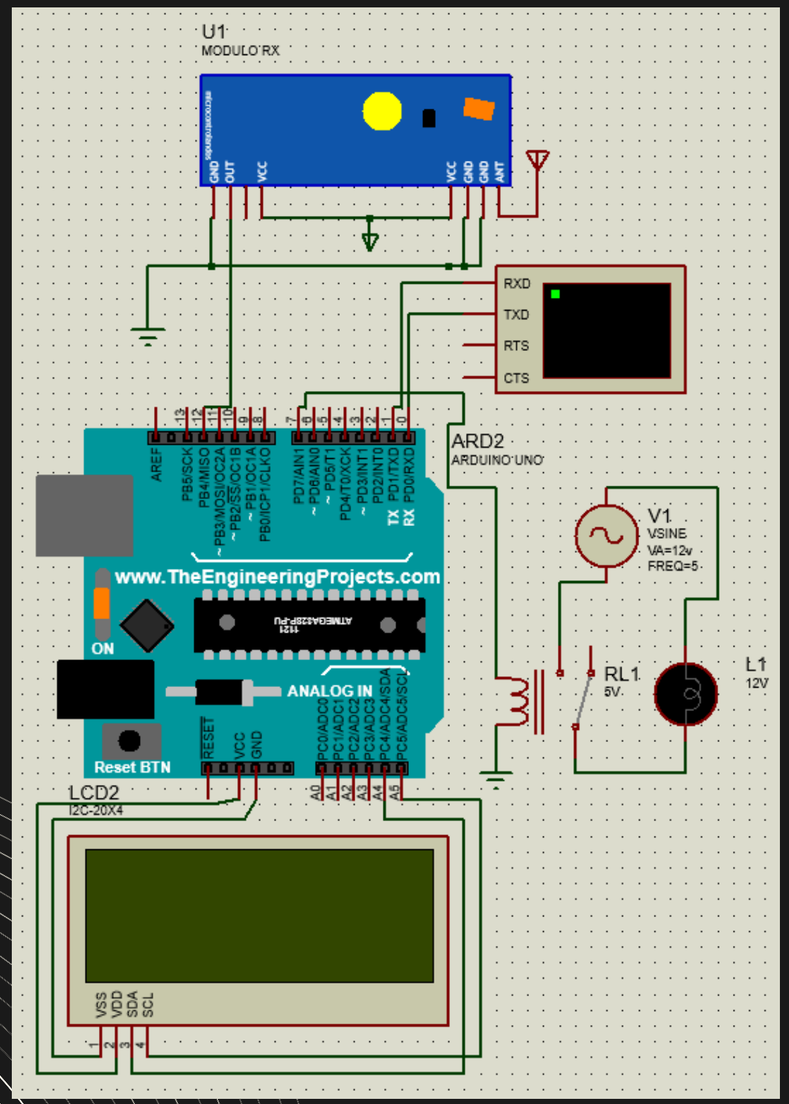
ARQUITECTURA TX



El riel de 5V del UNO alimenta al módulo RF, al sensor DHT11 y al back-light del LCD. La línea de GND es común a todos los módulos.
Comunicación RF (VirtualWire): Se utiliza un pin GPIO dedicado (DATA) para modular ASK a unos 4 kbps, definidos en el codigo.
Sensor DHT11 utiliza un solo hilo de datos bidireccional conectado al pin 4
Pantalla I²C 20×4 comparte el bus I²C del Arduino en los pines A4 (SDA) y A5 (SCL). Además de SDA y SCL, solo necesita alimentación (VCC) y tierra (GND).
Los pines DO (RXO) y D1 (TXO) del Arduino van a un terminal virtual en Proteus, donde se muestran los mensajes de Serial.print().

Arduino UNO	Señal	Módulo / Línea
5 V	VCC	RF-TX VCC, DHT11 VDD, LCD VDD
GND	GND	RF-TX GND, DHT11 GND, LCD GND
D12 (PB4)	DATA_RF	DATA del módulo VirtualWire (TX)
D2 (PD2)	DATA_DHT11	Línea de datos del sensor DHT11
A4 (PC4)	SDA	SDA del LCD I²C (LiquidCrystal_I2C)
A5 (PC5)	SCL	SCL del LCD I²C
D0 (PD0)	RX0	TXD del “terminal” Proteus (lo que vemos como monitor serial)
D1 (PD1)	TX0	RXD del “terminal” Proteus (lo que vemos como monitor serial)

ARQUITECTURA RX



El riel de 5V del UNO alimenta al módulo RF, al sensor DHT11 y al back-light del LCD. La línea de GND es común a todos los módulos.
módulo RX (“DATA OUT”) va a un GPIO del Arduino(PIN 12). Solo requiere líneas que DATA, VCC y GND.
Uno de los terminales de la bobina del relé se alimenta a 5 V; el otro va al pin digital elegido (PIN D7) pasando antes por un transistor de conmutación. El contacto del relé se inserta en serie entre la fuente de 12 V (Vsine) y la lámpara L1, de modo que al energizar la bobina ésta cierra el circuito de 12 V y enciende la lámpara.
Pantalla I²C 20×4 comparte el bus I²C del Arduino en los pines A4 (SDA) y A5 (SCL). Además de SDA y SCL, solo necesita alimentación (VCC) y tierra (GND).
Los pines D0 (RX0) y D1 (TX0) del Arduino van a un terminal virtual en Proteus, donde se muestran los mensajes de Serial.print().

Arduino UNO	Señal	Módulo / Línea
5 V	VCC	Módulo RX VCC, LCD VCC, bobina del relé (RL1)
GND	GND	Módulo RX GND, LCD GND, bobina del relé (RL1)
D12 (PB4)	DATA_RX	DATA OUT del módulo RX (VirtualWire)
D7 (PD7)	RELAY_CTRL	Control de la bobina del relé
A4 (PC4)	SDA	SDA del LCD I²C
A5 (PC5)	SCL	SCL del LCD I²C
D0 (PD0)	RX0	TXD del terminal virtual de Proteus (depuración serie)
D1 (PD1)	TX0	RXD del terminal virtual de Proteus (depuración serie)