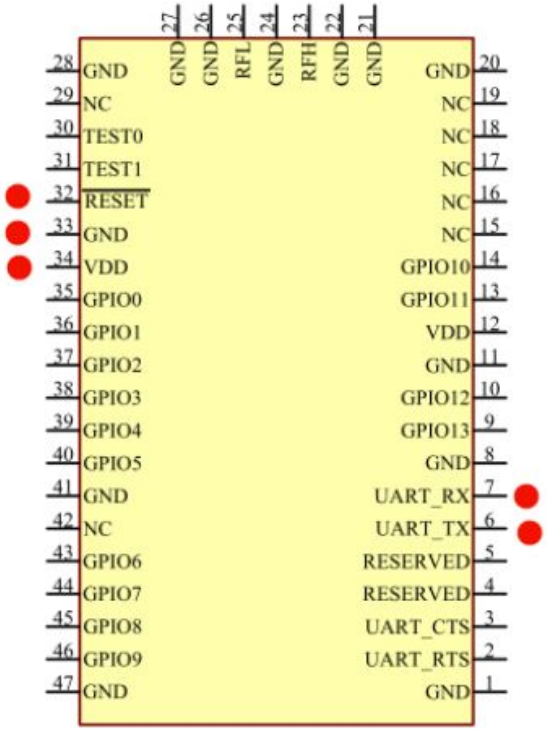
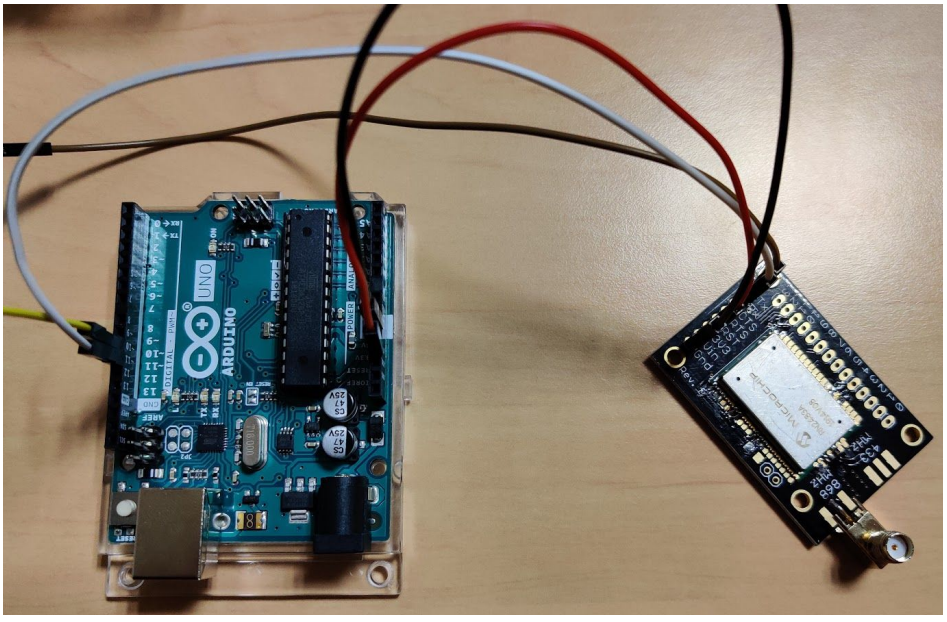
Step 1 :

Soudage du microship sur la carte
Besoin des pins TX, RX, RESET, 3V3, GND à trouver dans la datasheet du module



Step 2 :

Cablâge du montage :
Les pins TX et RX du module sont respectivement branchés sur les pins 11 et 10 de l'Arduino :



Step 3 :

Annex 1 is containing the code to send and receive packet over LoRa. This code is also displaying on serial monitor the module identifiers and each sent and received packet.

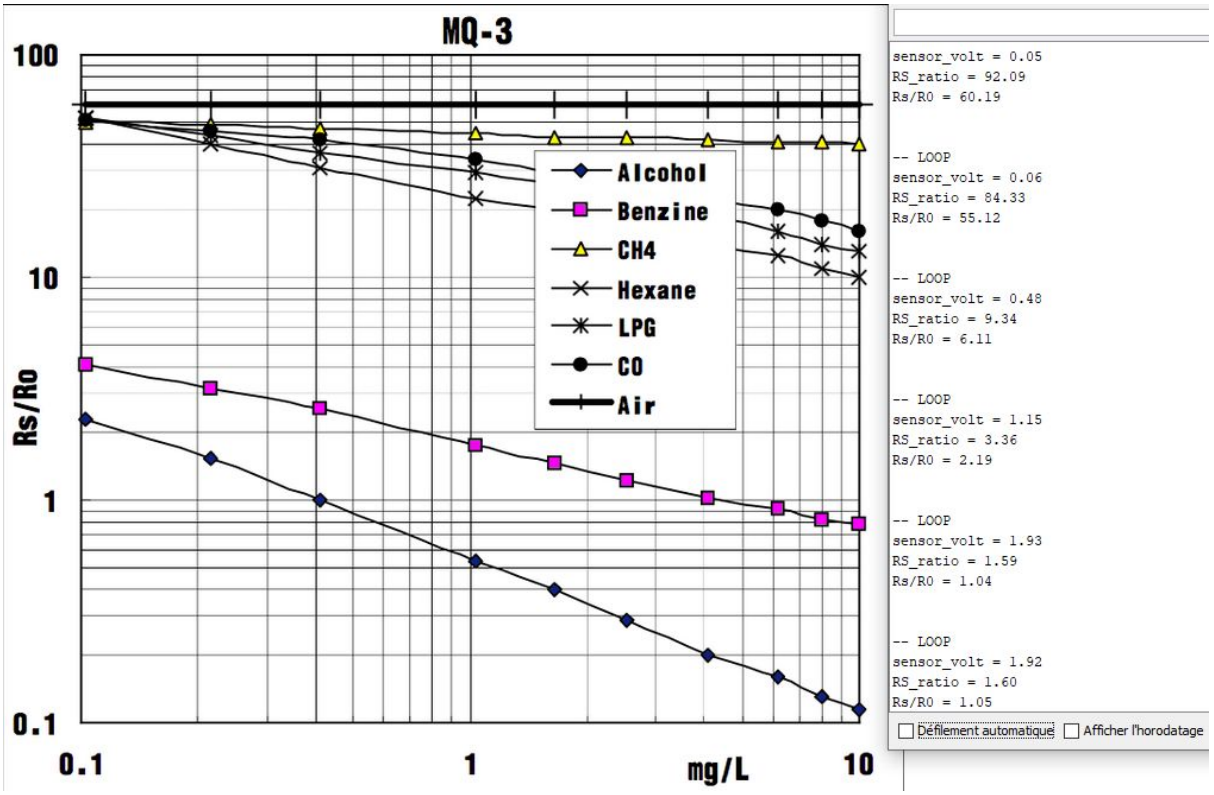
```
-- LOOP
Sending: mac tx uncnf 1 00
Successful transmission
-- LOOP
Sending: mac tx uncnf 1 00
Successful transmission
-- LOOP
Sending: mac tx uncnf 1 00
Successful transmission. Received 11223344
-- MESSAGE
Received 4 bytes on port 1: 17 34 51 68
-- LOOP
Sending: mac tx uncnf 1 00
Successful transmission
```

We can see on the serial monitor that the module is sending a “00” value each 10 seconds. On each message sent the module is checking if a message has been received. If we send a message from The Things Network webpage, we can see the received packet on the serial monitor as following :

Step 4 :
The method gas_variation_detection() displays the raw sensor value between 0 and 5V. This method can only detect a gas concentration variation in the environment.

The method gas_calibration() calibrates the sensor by measuring the resistivity value of the sensor in clean air. This value is used in the method gas_measurement().

The method gas_measurement() calculates the resistivity ratio between current gas and clean air values.
Comparing the value to the following graph, we can find the concentration of a particular gas in the environment.



On this test we put hydroalcoholic gel under the sensor. As we can observe on the monitor the ratio is dropping to around 1.04. Knowing that we are observing alcohol we can determine the concentration to be around 0.4mg/L.

Annex 1 : Code to send and receive data over LoRa

```
#include <TheThingsNetwork.h>
#include <rn2xx3.h>
#include <SoftwareSerial.h>

// Set your DevAddr, NwkSKey, AppSKey and the frequency plan
const char *devAddr = "26013B1D";
const char *nwkSKey = "227EC8A2C413CF2081E4C53333E95B48";
const char *appSKey = "ADC759AF56DBFB8DF302EE7B2CBC3D27";

#define debugSerial Serial
#define TX  10
#define RX  11

SoftwareSerial loraSerial(TX, RX);
// Replace REPLACE_ME with TTN_FP_EU868 or TTN_FP_US915
#define freqPlan TTN_FP_EU868

TheThingsNetwork ttn(loraSerial, debugSerial, freqPlan);

#define LED_PIN 13

void led_on()
{
  digitalWrite(LED_PIN, HIGH);
}

void led_off()
{
  digitalWrite(LED_PIN, LOW);
}

void message(const uint8_t *payload, size_t size, port_t port)
{
  debugSerial.println("-- MESSAGE");
  debugSerial.print("Received " + String(size) + " bytes on port " +
String(port) + ":");

  for (int i = 0; i < size; i++)
  {
      debugSerial.print(" " + String(payload[i]));
  }

  debugSerial.println();
```

```
  if (payload[0] == 1){led_on();} else {led_off();}
}

void setup()
{

  pinMode(LED_PIN, OUTPUT);

  loraSerial.begin(9600);
  debugSerial.begin(9600);

  //recalculation of lora module baudrate
  String data = "";
  debugSerial.println("dem");
  for(int i = 0; i<10 && data ==""; i++)
  {
      loraSerial.write((byte)0x00);
      loraSerial.write(0x55);
      loraSerial.println();
      loraSerial.println("sys get ver");
      data = loraSerial.readStringUntil('\n');
      delay(1000);
  }
  debugSerial.println("fin dem");

  debugSerial.println("-- PERSONALIZE");
  ttn.personalize(devAddr, nwkSKey, appSKey);

  //For downlink
  ttn.onMessage(message);

  debugSerial.println("-- STATUS");
  ttn.showStatus();
}

void loop()
{
  debugSerial.println("-- LOOP");

  // Prepare payload of 1 byte to indicate LED status
  byte payload[1];
  payload[0] = (digitalRead(LED_BUILTIN) == HIGH) ? 1 : 0;

  // Send it off
  ttn.sendBytes(payload, sizeof(payload));
```

```
  delay(10000);
}
```

▲	09:25:11	21	1	payload: 00
▲	09:24:59	20	1	payload: 00
▲	09:24:47	19	1	payload: 00
▼	09:24:35		1	payload: 11 22 33 44
▲	09:24:34	18	1	payload: 00