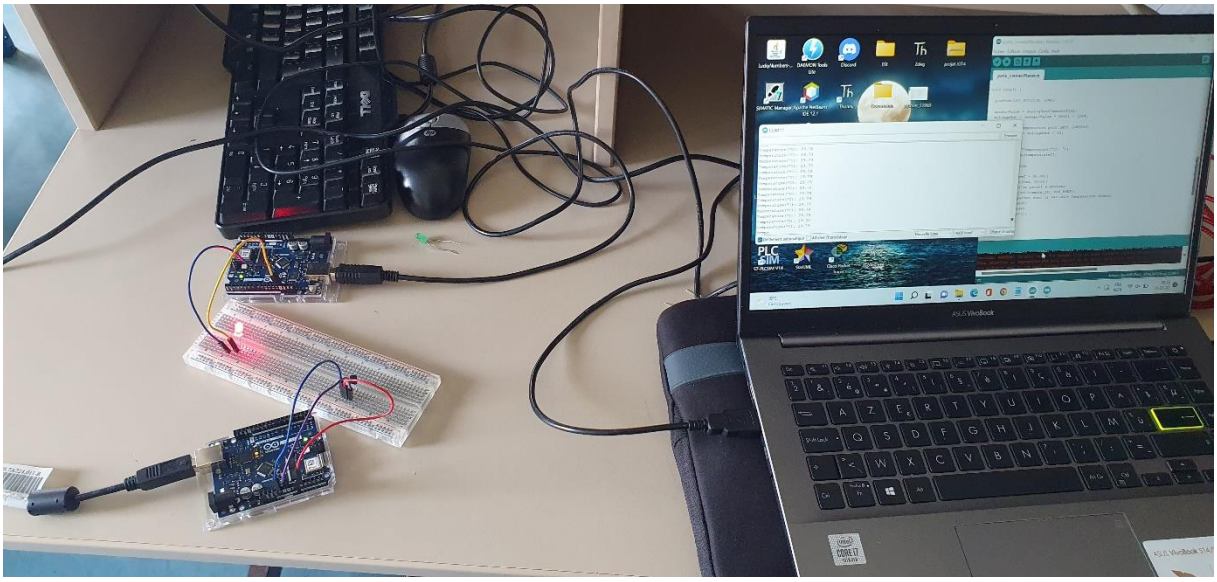
Sabouri Abdessamad 53960
Mawuvi Koffi 54309

13/05/2022

RAPPORT PROJET IOT4



But :

Dans notre cas on va utiliser deux « Arduino uno wifi rev2 » avec une connexion udp. (L'UDP (User Datagram Protocol) est un protocole sans connexion de la suite des protocoles Internet qui travaille au niveau de la couche transport et qui a été défini en 1980 dans la RFC (Request for Comments) 768. En tant qu'alternative au TCP fonctionnant de façon plus simple et quasiment sans retard, l'UDP est utilisé pour la transmission rapide de paquets de données dans des réseaux IP. Les domaines d'application typiques de l'UDP sont donc les requêtes DNS, les connexions VPN et le streaming audio et vidéo.) **sur le premier Arduino on va mettre un capteur de température LM35 D, et sur le deuxième Arduino on va mettre une LED, en utilisant une condition pour le capteur de température (Dès qu'il arrive à une certes température il va activer la Led dans l'autre Arduino grâce au transfert des paquets).**

Considérons un cas précis : l'Arduino #1 communique avec l'Arduino #2. Il existe de nombreuses méthodes pour permettre la communication entre deux Arduino. Nous pouvons en choisir un en fonction de la portée de communication. Le tableau ci-dessous montre certaines méthodes et leur portée de communication.

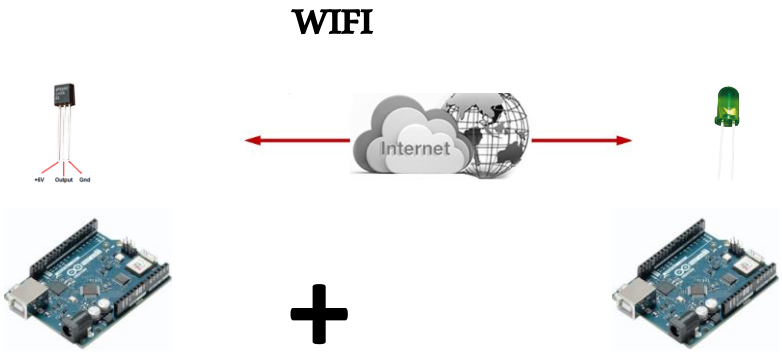
Méthodes	Varier
I2C	très court
IPS	très court
UART (TTL)	très court
UART (RS-232/485/422)	court
Bluetooth	court
LoRa	longue
Ethernet/Wi-Fi	illimité (*)

Introduction :

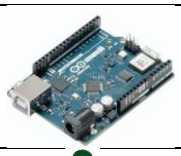
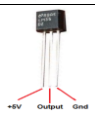
Nous allons réaliser ce projet ci-dessous :

Sabouri Abdessamad 53960
Mawuvi Koffi 54309

13/05/2022



Matériel :

	2 x Arduino Uno Wifi rev2
	Led
	Capteur de température LM35 dz
	Breadboard

Installation de l'environnement de travail :

Pour contrôler le Node, nous avons dû installer un IDE (environnement de développement). L'IDE que nous utilisons est totalement gratuit (programme open source) lien disponible ci-dessous. https://www.arduino.cc/en/software Une fois l'installation d'IDE réalisée, nous pouvons engagez la deuxième étape qui consiste à rajouter une librairie spéciale qui contient une liste gestionnaire de carte Prédéfini, notamment celui de notre microcontrôleur : Arduino uno wifi rev2

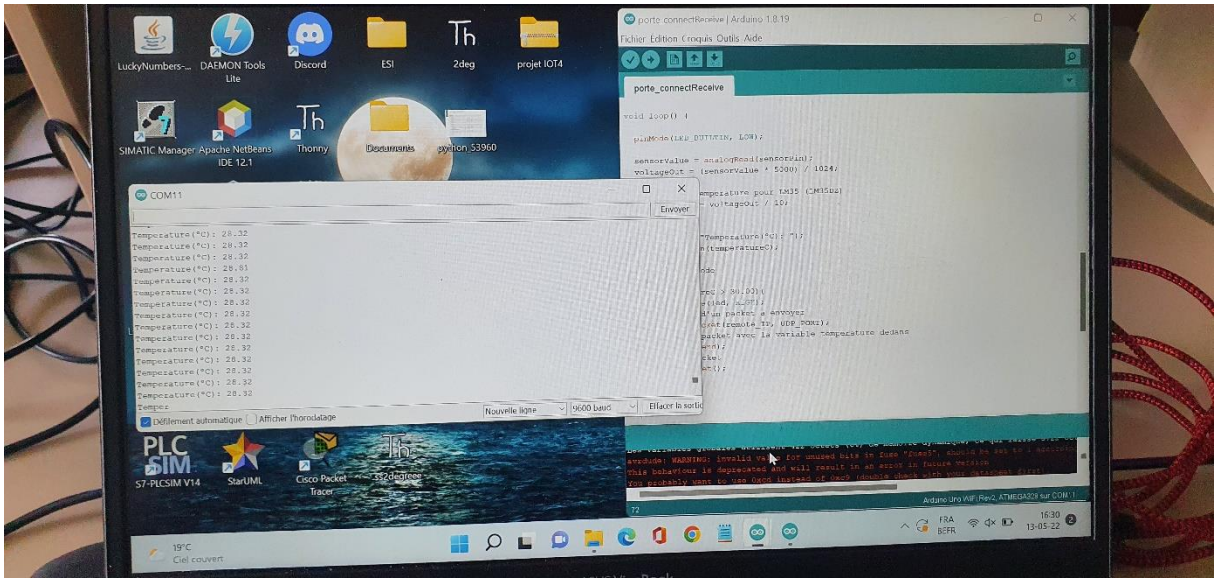
Sabouri Abdessamad 53960
Mawuvi Koffi 54309

13/05/2022

Le capteur température LM35 dz :

Ce capteur de température permet d'acquérir une température ambiante.

Une fois alimenté, il va délivrer une tension analogique proportionnelle à la température.
Tension d'alimentation : 4V à 30V
Etendu de mesure : 0°C à 100°C
Précision : ±0,75 °C (typique)
Echelle : 10mV/°C
Calibration : 0mV à 0°C, 1000mV à 100°C
Ce produit est facilement utilisable à travers un pin analogique d'une carte Arduino.



Branchement :

La connexion d'un **LM35** à l'**Arduino** est très simple car il vous suffit de connecter 3 broches. Commencez par connecter la broche + V_S à la sortie 5 V de l'**Arduino** et la broche GND à la terre. Ensuite, connectez la broche du milieu (V_{OUT}) à l'une des entrées analogiques de l'Arduino.

Code :

1. Cette première partie du code c'est le rôle du RECEIVE celui qui va recevoir les paquets :

Sabouri Abdessamad 53960
Mawuvi Koffi 54309

13/05/2022

```
char ssid[] = "Mywifi";
char pass[] = "12345678";
WiFiUDP Udp;
IPAddress local_IP(192,168,4,1);
IPAddress gateway(192,168,4,1);
IPAddress subnet(255,255,255,0);
int status = WL_IDLE_STATUS;
#define UDP_PORT 4210

// UDP Buffer
char packetBuffer[UDP_TX_PACKET_MAX_SIZE];

boolean alreadyConnected = false;
void setup()
{
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.println("Access Point");

  //CONFIG WIFI
  Serial.print("Setting soft-AP configuration ... ");
  status = WiFi.beginAP(ssid, pass);
  Serial.println(WiFi.localIP());

  if (status != WL_AP_LISTENING) {
    Serial.println("Creating access point failed");
    // don't continue
    while (true);
  }
  delay(10000);
  printWiFiStatus();

  //set output for door
  pinMode(13, OUTPUT);

   // Begin listening to UDP port
  Udp.begin(UDP_PORT);

 Serial.print("Listening on UDP port ");
 Serial.println(UDP_PORT);
}


void loop()
{
 if (status == WL_AP_CONNECTED) {
     // a device has connected to the AP
     Serial.println("Device connected to AP");
   } else {
     // a device has disconnected from the AP, and we are back in listening mode
     Serial.println("Device disconnected from AP");
   }
 Udp.parsePacket();
 //lire le packet recu
 Serial.println("d");
 Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
 //et avec le packet tu fais tes bails
 if (packetBuffer[0]){
   Serial.println("high");
   digitalWrite(13, HIGH);
   delay(7000);
   digitalWrite(13, LOW);
   delay(1);
   packetBuffer[0] = 0;
 }
}

void printWiFiStatus() {
 // print the SSID of the network you're attached to:
 Serial.print("SSID: ");
 Serial.println(WiFi.SSID());

 // print your WiFi shield's IP address:
 IPAddress ip = WiFi.localIP();


  // print where to go in a browser:
  Serial.print("To see this page in action, open a browser to http://");
  Serial.println(ip);

}
```

Sabouri Abdessamad 53960
Mawuvi Koffi 54309

13/05/2022

2. Cette deuxième partie du code c'est le rôle du SEND celui qui va créer le wifi et envoyer les paquets :

```
//Library
//#include <SPI.h>
#include <WiFiNINA.h>
#include <WiFiUdp.h>


//variableWifi
char ssid[] = "Mywifi";
char pass[] = "12345678";
WiFiUDP Udp;
IPAddress remote_IP(192,168,4,1);
#define UDP_PORT 4210
int status = WL_IDLE_STATUS;
char send = 'y';


const int sensorPin = A0;
int led = 13;
float sensorValue;
float voltageOut;

float temperatureC;
float temperatureF;


void setup() {
  Serial.begin(9600);



//wifi
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid, pass);
```

```
    // wait 10 seconds for connection:
    delay(10000);
  }

  // you're connected now, so print out the status:
  printWifiStatus();

  // Begin UDP port
  Udp.begin(UDP_PORT);
  Serial.print("Opening UDP port ");
  Serial.println(UDP_PORT);

  pinMode(sensorPin, INPUT);
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}



void loop() {

  pinMode(LED_BUILTIN, LOW);

  sensorValue = analogRead(sensorPin);
  voltageOut = (sensorValue * 5000) / 1024;

    // calcul temperature pour LM35 (LM35DZ)
  temperatureC = voltageOut / 10;


  Serial.print("Temperature(°C): ");
  Serial.println(temperatureC);

  //lié a mon code

  if (temperatureC > 30.00){
    digitalWrite(led, HIGH);
    //creation d'un packet a envoyer
    Udp.beginPacket(remote_IP, UDP_PORT);
    //ecrit le packet avec la variable temperature dedans
    Udp.write(send);
    //fin du packet
    Udp.endPacket();
    delay(100);
    }
  else
  {
    digitalWrite(led, LOW);
  }


}



//wifi statut
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your board's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);


   // print the received signal strength:
   long rssi = WiFi.RSSI();
   Serial.print("signal strength (RSSI):");
   Serial.print(rssi);
   Serial.println(" dBm");
 }
```

Sabouri Abdessamad 53960
Mawuvi Koffi 54309

13/05/2022

Ci-dessous le lien pour voir le résultat final du projet :

Vidéo du résultat

Fin.