

Nom :

Prénom :

TP SIN
Etude d'un projet
Groupe de deux élèves

Support : Chocolaterie Elot

(Vous réaliserez tous les schémas Sysml sur logiciel, puis vous les imprimerez et les rajouterez au dossier)

Pré requis (l'élève doit savoir):

- Savoir utiliser un ordinateur
- Savoir réaliser un schéma Sysml

Programme

Objectif terminale :

L'élève doit être capable de faire l'étude d'un projet

Matériels :

- Logiciel Ganttproject
- Vidéo chocolaterie Elot

1. **Cahier des charges**

Le maître chocolatier veut pouvoir pour une question de sécurité d'hygiène alimentaire, lire sur son téléphone portable ou depuis un ordinateur la température du frigo en temps réel, accéder aux mesures de la journée. Il veut pouvoir aussi afficher des informations de fonctionnement sur un afficheur (connexion du client, envoi de la valeur, affichage de la température).

2. **Contraintes**

- Coût financier sans la main d'œuvre : 500 E Max
- Equipe de deux personnes avec un chef de projet
- Toutes les personnes sont susceptibles de faire toutes les tâches.
- Temps alloué pour le projet : le trimestre
- Le professeur joue le rôle du client et du chef d'entreprise pour lequel vous travaillez,
- **Attention le montage se trouvera sur une chaîne de fabrication alimentaire**
- **Réduire l'impact environnemental au maximum**

3. **Travail demandé pour le dossier**

- **Dans un premier temps vous travaillerez en groupe, après chacun avec ses tâches (cf suite détaillé en vert)**
- Travail en groupe
- Traduire le cahier des charges sous forme de diagramme des exigences
 - TP d'aide

Nom :

Prénom :

- SysML (MagicDraw)
- Diagramme état transition (Automgen)



- Réalisation d'une représentation en sysml (balance de ménage)
- Réalisation d'une représentation en sysml (serrure biométrique)

Pour la réalisation sur modelio (voir les vidéos)

VIDÉO

Vidéo (schémas sysml Modelio)

<http://sti2dsinhyome.fr/sommairevideosysml.html>

- En questionnant le client, trouver les différentes possibilités de matériel pour répondre au cahier des charges
- Dans un tableau lister les idées pour chaque exigence et indiquer leurs avantages et inconvénients puis en choisir une

Idées choisies	Avantage	Inconvénient

Nom :

Prénom :

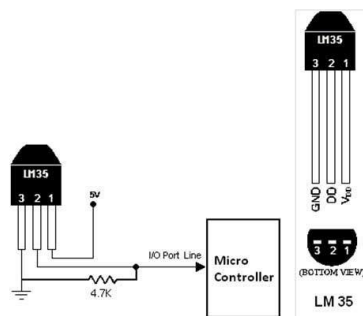
- Etude technique

- Pour la suite du projet on prendra les éléments suivants :
 - Téléphone Android
 - Sonde de température LM35
 - Ampli Op LM358N
 - Convertisseur analogique numérique 8 bits ADC0801
 - Circuit intégré liaison I2C PCF8574
 - Circuit intégré 555 pour la temporisation
 - Afficheur LCD 2x16 I2C DFR0063
 - Réseau interne de de la société
 - Base de données Mysql
 - Platine d'essai
 - Carte Arduino Méga ou Uno avec shield internet
 - Application php et Arduino pour gérer le système

- Présentation des éléments

- Branchement : LM35 et caractéristiques

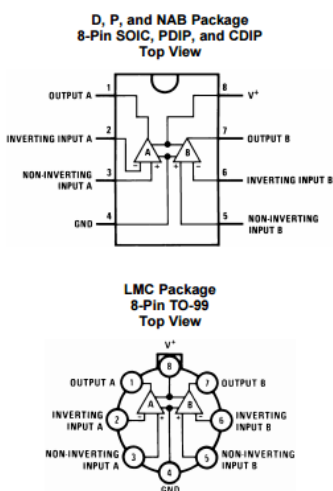
<http://www.ti.com/lit/ds/snis159g/snis159g.pdf>



Ce capteur de température linéaire LM35 peut être utilisé pour mesurer la température ambiante. Fonctionnant de -40 à 150 degrés Celsius, sa sensibilité est de 10mV par degré et sa tension de sortie est proportionnelle à la température.

- Branchement : LM 358N et caractéristiques

<http://www.ti.com/lit/ds/symlink/lm158-n.pdf>



PIN				Pin Functions	
D/P/LMC NO.	DSBGA NO.	NAME	TYPE	DESCRIPTION	
1	A1	OUTA	O	Output, Channel A	
2	B1	-INA	I	Inverting Input, Channel A	
3	C1	+INA	I	Non-Inverting Input, Channel A	
4	C2	GND / V-	P	Ground for Single supply configurations. negative supply for dual supply configurations	
5	C3	+INB	I	Output, Channel B	
6	B3	-INB	I	Inverting Input, Channel B	
7	A3	OUTB	O	Non-Inverting Input, Channel B	
8	A2	V+	P	Positive Supply	

Copyright © 2000–2014, Texas Instruments Incorporated

[Submit Documentation Feedback](#)

1 Features

- Available in 8-Bump DSBGA Chip-Sized Package, (See AN-1112, [SNVA009](#))
- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100 dB
- Wide Bandwidth (Unity Gain): 1 MHz (Temperature Compensated)
- Wide Power Supply Range:
 - Single Supply: 3V to 32V
 - Or Dual Supplies: $\pm 1.5V$ to $\pm 16V$
- Very Low Supply Current Drain (500 μA)—Essentially Independent of Supply Voltage
- Low Input Offset Voltage: 2 mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Large Output Voltage Swing
- Unique Characteristics:
 - In the Linear Mode the Input Common-Mode Voltage Range Includes Ground and the Output Voltage Can Also Swing to Ground, even though Operated from Only a Single Power Supply Voltage.
 - The Unity Gain Cross Frequency is Temperature Compensated.
 - The Input Bias Current is also Temperature Compensated.
- Advantages:
 - Two Internally Compensated Op Amps
 - Eliminates Need for Dual Supplies
 - Allows Direct Sensing Near GND and V_{OUT} Also Goes to GND
 - Compatible with All Forms of Logic
 - Power Drain Suitable for Battery Operation

LM158-N, LM258-N, LM2904-N, LM358-N

SNOSBT3I – JANUARY 2000 – REVISED DECEMBER 2014

www.ti.com

Electrical Characteristics: LM358, LM2904 (continued)

$V^+ = +5.0 V$, See⁽¹⁾, unless otherwise stated

PARAMETER		TEST CONDITIONS		LM358			LM2904			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
Input Common-Mode Voltage Range		V ⁺ = 30 V, See ⁽⁴⁾ (LM2904, V ⁺ = 26 V)		0		V ⁺ -2	0		V ⁺ -2	V
Large Signal Voltage Gain		V ⁺ = +15 V		15			15			V/mV
		(V _O = 1 V to 11 V)								
		R _L ≥ 2 kΩ								
Output	V _{OH}	V ⁺ = 30 V	R _L = 2 kΩ	26			22			V
Voltage		(LM2904, V ⁺ = 26 V)	R _L = 10 kΩ	27	28		23	24		V
Swing	V _{OL}	V ⁺ = 5 V, R _L = 10 kΩ		5 20			5 100			mV
Output Current	Source	V _{IN} ⁺ = 1 V, V _{IN} ⁻ = 0 V,		10 20			10 20			mA
		V ⁺ = 15 V, V _O = 2 V								
	Sink	V _{IN} ⁻ = 1 V, V _{IN} ⁺ = 0 V,		5 8			5 8			mA
		V ⁺ = 15 V, V _O = 2 V								

Nom :
○ Afficheur LCD I2C DRF0063

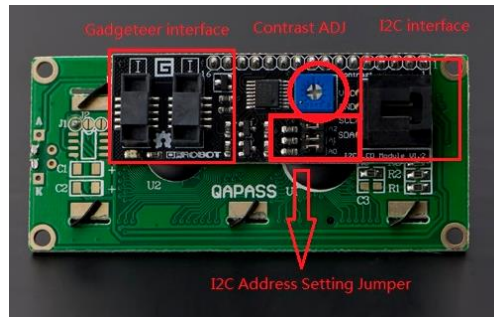
Prénom :

[https://www.dfrobot.com/wiki/index.php/I2C/TWI_LCD1602_Module_\(Gadgeteer_Compatible\)_ \(SKU: DFR0063\)](https://www.dfrobot.com/wiki/index.php/I2C/TWI_LCD1602_Module_(Gadgeteer_Compatible)_ (SKU: DFR0063))



Specification

- I2C Address: 0x20-0x27 (0x20 default)
- Back lit (Blue with white char color)
- Supply voltage: 5V
- Interface: I2C/TWI x1, Gadgeteer interface x2
- Adjustable contrast
- Size: 82x35x18 mm

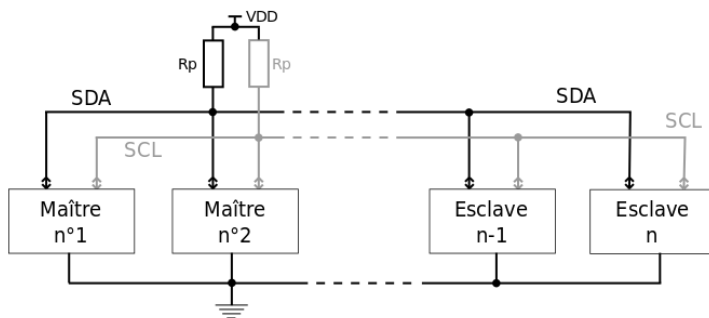


I²C est un bus série synchrone bidirectionnel half-duplex (voir cours liaison série), où plusieurs équipements, maîtres ou esclaves, peuvent être connectés au bus.

Les échanges ont toujours lieu entre un seul maître et un (ou tous les) esclave(s), toujours à l'initiative du maître (jamais de maître à maître ou d'esclave à esclave). Cependant, rien n'empêche un composant de passer du statut de maître à esclave et réciproquement.

La connexion est réalisée par l'intermédiaire de deux lignes :

- SDA (Serial Data Line) : ligne de données bidirectionnelle,
- SCL (Serial Clock Line) : ligne d'horloge de synchronisation bidirectionnelle.



Address Setting

A2	A1	A0	IIC Address
0	0	0	0x20
0	0	1	0x21
0	1	0	0x22
0	1	1	0x23
1	0	0	0x24
1	0	1	0x25
1	1	0	0x26
1	1	1	0x27

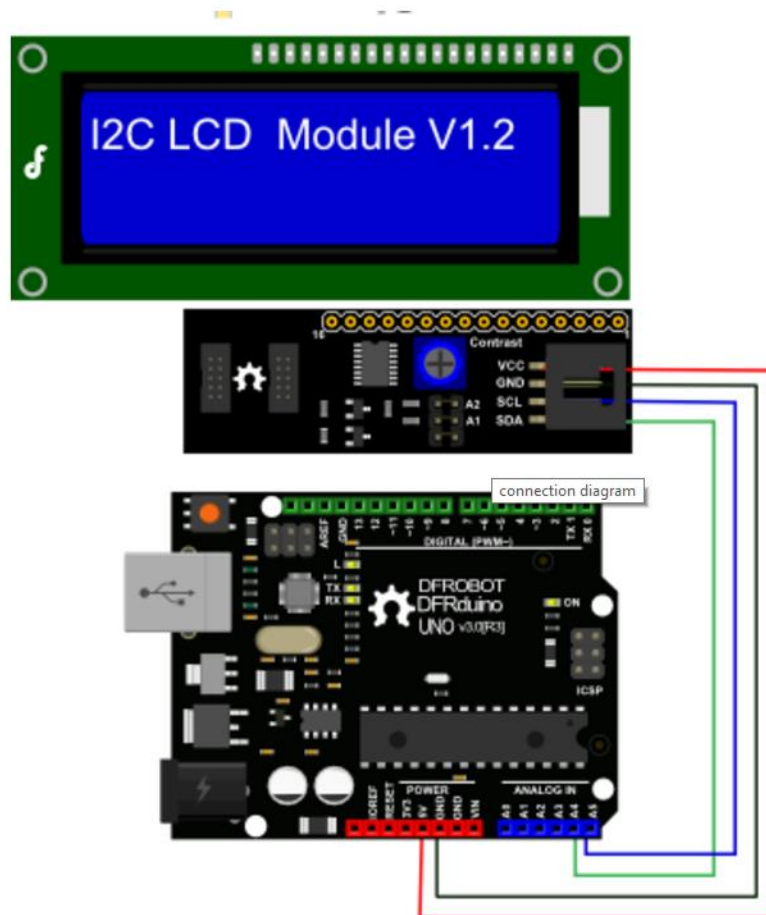
- 0: The Jumper Cap is connected
- 1: The Jumper Cap is disconnected

NOTE: The default address is 0x20. All the jumper caps will be connected from the factory.

Nom :

Prénom :

Connection



NOTE: V1.2 has a different power pinout from V1.1, please check the history version for the old connection diagram.

Arduino UNO: connect SDA to pin A4 and SCL to pin A5 on your Arduino.

Arduino Leonardo: connect SDA to digital pin 2 and SCL to digital pin 3 on your Arduino.

Arduino Méga: connect SDA to digital pin 20 and SCL to digital pin 21 on your Arduino.

Library Support Functions

- **LiquidCrystal_I2C()** //set the LCD address for a 16 chars and 2 line display
- **init()** //Initialization for the LCD
- **clear()** //clear display, set cursor position to zero
- **home()** //set cursor position to zero
- **createChar()** //Fill the first 8 CGRAM locations with custom characters
- **setCursor()** //set the position of the cursor
- **cursor()** //Turns the underline cursor on
- **noCursor()** //Turns the underline cursor off
- **blink()** //Turn on the blinking cursor
- **noBlink()** //Turn off the blinking cursor
- **display()** //Turn the display on(quickly)
- **noDisplay()** //Turn the display Off(quickly)
- **backlight()** //Turn the backlight on
- **noBacklight()** //Turn the backlight off
- **scrollDisplayLeft()** //Make the display scroll left without changing the RAM
- **scrollDisplayRight()** //Make the display scroll right without changing the RAM
- **autoscroll()** //This will 'right justify' text from the cursor
- **noAutoscroll()** //This will 'left justify' text from the cursor
- **leftToRight()** //This is for text that flows Left to Right
- **rightToLeft()** //This is for text that flows Right to Left

```
LiquidCrystal_I2C lcd(0x20,16,2); // set the LCD address to 0x20 for a 16 chars and 2 line display
```

Nom :

Prénom :

test | Arduino 1.8.1

Fichier Édition Croquis Outils Aide



test

```
/*
Analog Input
Demonstrates analog input by
turning on and off a light
The amount of time the LED w
the value obtained by analog

The circuit:
* Potentiometer attached to
* center pin of the potenti
* one side pin (either one)
* the other side pin to +5V
* LED anode (long leg) attac
* LED cathode (short leg) at

* Note: because most Arduino
to pin 13 on the board, the
```

Gestionnaire de bibliothèque

Type

Tout

Sujet

Tout

liquid

LcdProgressBarDouble by Wilfried Loche

A library to draw some customized progress bar on any LCD display. Depends on LiquidChrystal library.

[More info](#)

LiquidCrystal I2C by Frank de Brabander Version 1.1.2 **INSTALLED**

A library for I2C LCD displays. The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.

[More info](#)

LiquidCrystal_PCF8574 by Matthias Hertel

A library for driving LiquidCrystal displays (LCD) by using the I2C bus and an PCF8574 I2C adapter. This library is derived from the original Arduino LiquidCrystal library and uses the original Wire library for communication.

Montage Proteus

regulateur à découpage ajustable - Proteus 8 Professional - Saisie de schéma

Fichier Edit Affichage Outils Projet Graphes Débogage Bibliothèque Gabarit Système Aide

Saisie de schéma

Circuit imprimé

Prendre des composants

Mots clés:

lcd

Tous les mots identiques?

Seuls composants modélisés?

Catégorie:

ACTIVE

04022D104KA12A

748

1210-101K

1458

Capacitor

CONNH2

DIODE

KT101

KT101-10X

LM35

LM317TEMP

MINRES10K

MINRES50K

POT-10K

SWITCH

Résultats (108):

Composant	Bibliothèque	Fabricant	Description
4054	CMOS		Quad Level Shifters/LCD Drivers With Input Latches
4055	CMOS		BCD To 7-Segment Decoder/LCD Driver
74HC4543	74HC		BCD-to-7 Segment Latch/Decoder/Driver for LCDs
74HC4543 IEC	74HC		BCD-to-7 Segment Latch/Decoder/Driver for LCDs
AGM12320	DISPLAY	AZ Displays	128x32 Graphical LCD with SED1520 controllers
AMP1RE120X54	DISPLAY		128x64 Graphical LCD with KS0108 controllers
AY0438/L	MICROCHIP	Microchip	32-Segment CMOS LCD Driver
AY0438/P	MICROCHIP	Microchip	32-Segment CMOS LCD Driver
EAD055102N-6	DISPLAY	ELECTRONIC ASSEMBLY	DOGS Graphics Series 102x64 Dots
EPH110254	DISPLAY		180x64 Graphical LCD with 3 x KS0108 controllers
EW12A03GLY	DISPLAY		128x32 Graphical LCD with SED1520 controllers
HD44780	DISPLAY		Industry standard alphanumeric LCD controller
HDI12864F-3	DISPLAY	HANTRONIX	128x64 Graphical LCD with SED1565 controller. Serial data input
HDI12864F-4	DISPLAY	HANTRONIX	128x64 Graphical LCD with SED1565 controller. Parallel data input
HDI12864L-4	DISPLAY	HANTRONIX	128x64 Graphical LCD with SED1565 controller. Parallel data input, LED Backlight
HDI12864L-6	DISPLAY	HANTRONIX	128x64 Graphical LCD with SED1565 controller. Selectable Interface, LED Backlight
HDM32G512B	DISPLAY	HANTRONIX	128x32 Graphical LCD with SED1520 controllers, LED Backlight
HDM32G512Y-3	DISPLAY	HANTRONIX	128x32 Graphical LCD with SED1520 controllers, Selectable Interface, VAC LED Backlight
IC2541	DISPLAY		128x64 Graphical TFT-LCD with Backlight
IC2541S I2C	DISPLAY		128x64 Graphical TFT-LCD with Backlight
KS0108	DISPLAY		Graphical LCD controller
LCDSTK502	DISPLAY	Amel	Amel STK-502 Custom LCD Display
LCDSTK504	DISPLAY	Amel	Amel STK-504 Custom LCD Display
LM12841BS1R	DISPLAY		128x64 Graphical LCD with KS0108 controllers

Aperçu JHD-2X16-I2C:

VSM DLL Model [LCDALPHA]

VDD

SCL

SDA

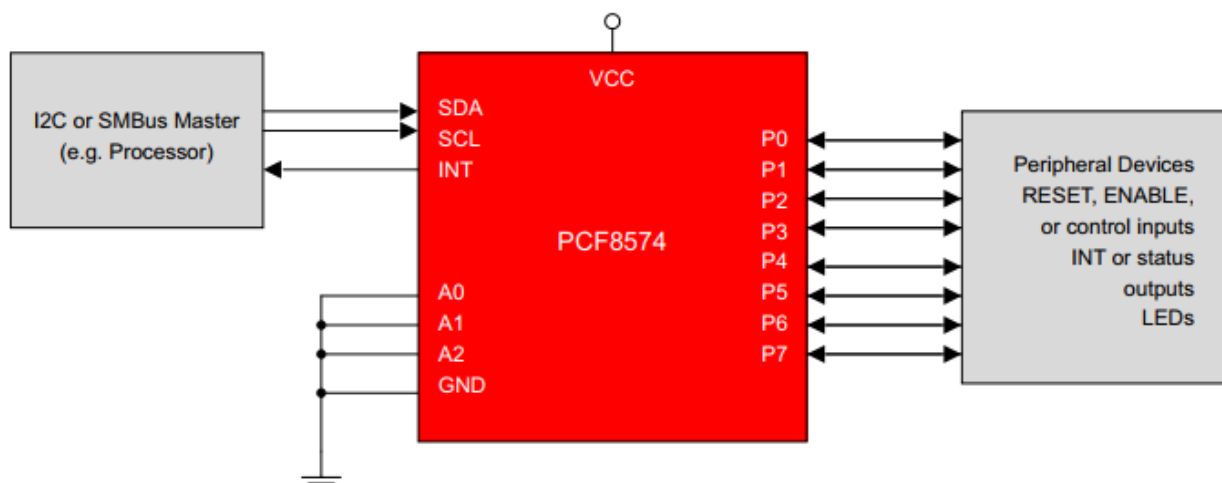
VSS

Aperçu circuit:

○ PCF8574

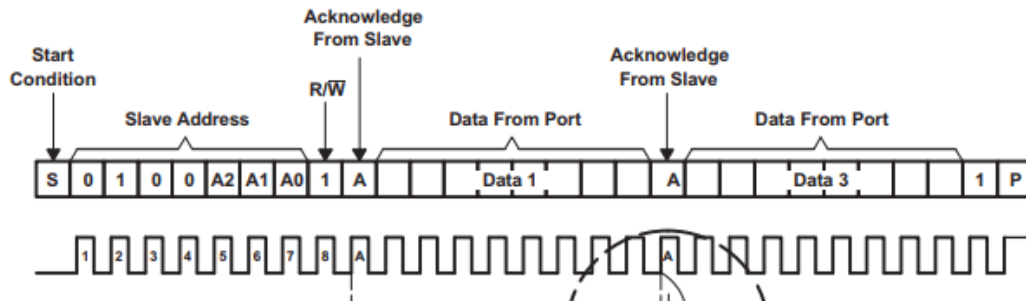
<http://www.ti.com/lit/ds/symlink/pcf8574.pdf>

C'est un circuit intégré qui permet de transformer une liaison parallèle 8 bits en liaison série i2c. Son adresse est modifiable en alimentant les bornes A0, A1, A2 en 5V ou 0V



Nom :
Trame :

Prénom :

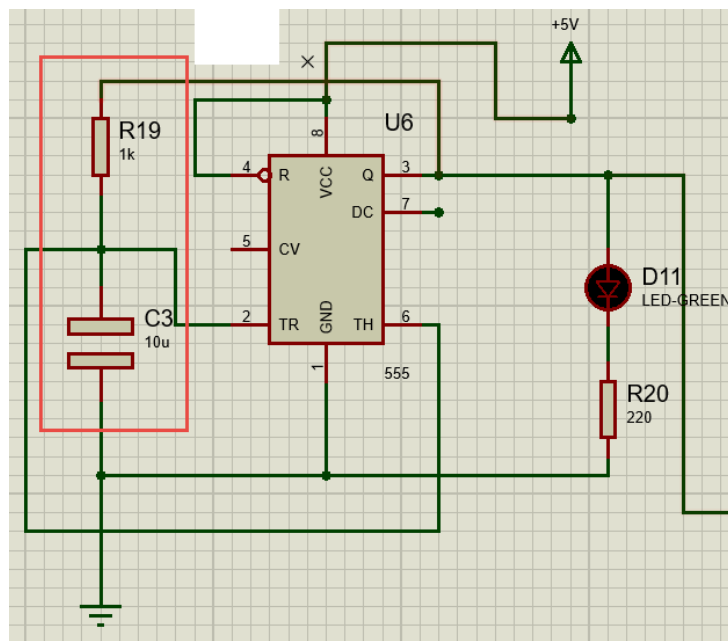


Exemple : En branchant A0 sur la masse et A1, A2 sur 5V, on obtient l'adresse suivante 0100110 ce qui fait en hexa 26

- NE555

C'est un circuit intégré qui permet de fournir une temporisation. On peut le faire fonctionner suivant 3 modes : monostable, astable ou bistable.

Dans notre TP nous allons l'utiliser en mode monostable. C'est-à-dire que ce montage permet de générer une impulsion d'une durée définie seulement à l'aide d'une résistance et d'un condensateur.



- Décomposition du projet en "tâches" et "sous-tâches" et planification du projet (réaliser le diagramme de Gantt sur Gantt Project)

Voir vidéo (fonctionnement Gantt Project)

<http://sti2dsinhyrome.fr/video%20diagrammegantt.html>

Attribution de ces tâches (qui fait quoi), quand et sur combien de temps se dérouleront les différentes tâches, quelles sont les opérations pouvant être menées en simultanée, quelles sont les marges de manœuvre en temps, en coûts etc. (§ Pert, Gantt & Cie)

- Présenter le système retenu sous forme de diagramme de cas d'utilisation

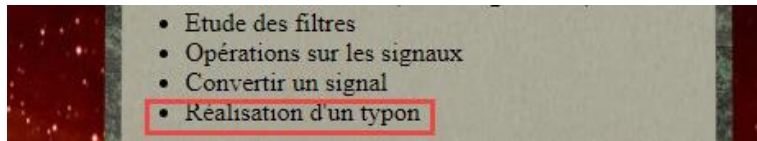
Nom :

Prénom :

- Etude structurelle :

- Branchement sonde sur carte Arduino (élève 1)

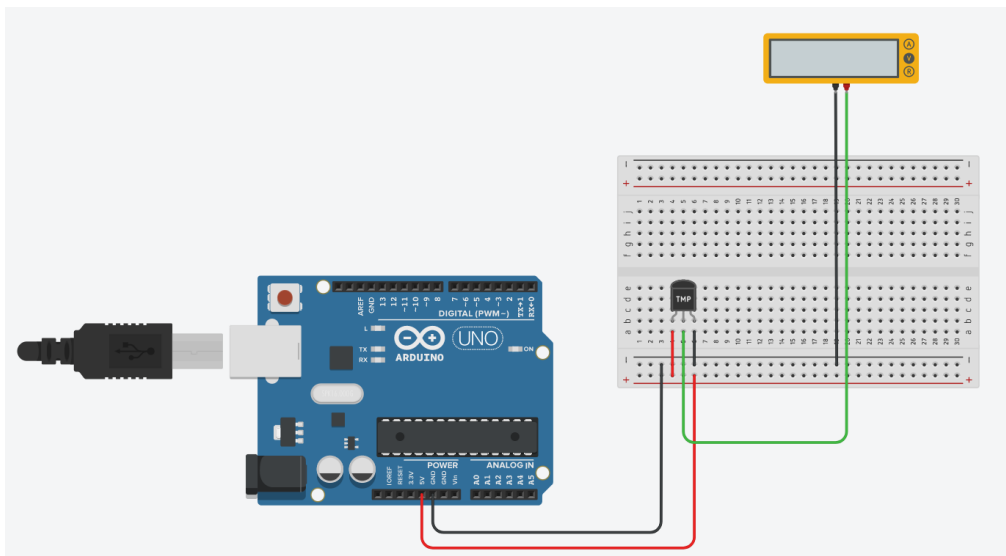
- TP d'aide



- Réaliser le schéma sur Proteus et les calculs du branchement du la sonde sur la carte Arduino. La température de la salle de la chaîne de fabrication ne dépasse pas 30 °C. La tension du signal arrivant à la carte doit varier de 0 à 5V pour une température de 0 à 50 °C.

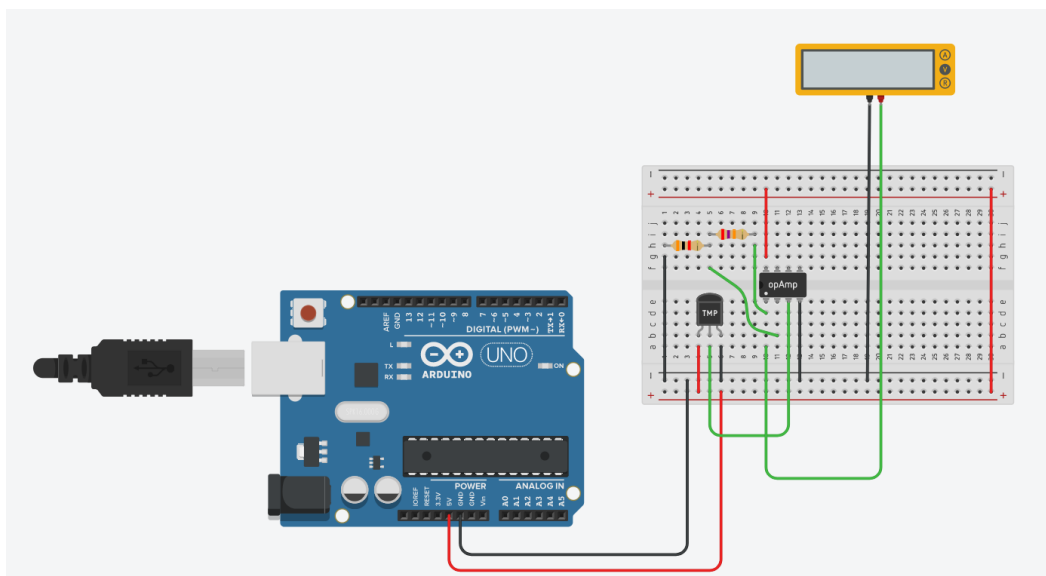
- Démarche :

- Une fois que vous avez branchez la sonde de température correctement, contrôler la tension de sortie et comparez-la à la température de la salle (voir ci-dessus sensibilité capteur)



- Une fois la valeur mesurée, rajouter le montage correspondant (voir cours opération sur les signaux) afin que pour 20°C on ait 2 V en sortie du montage.

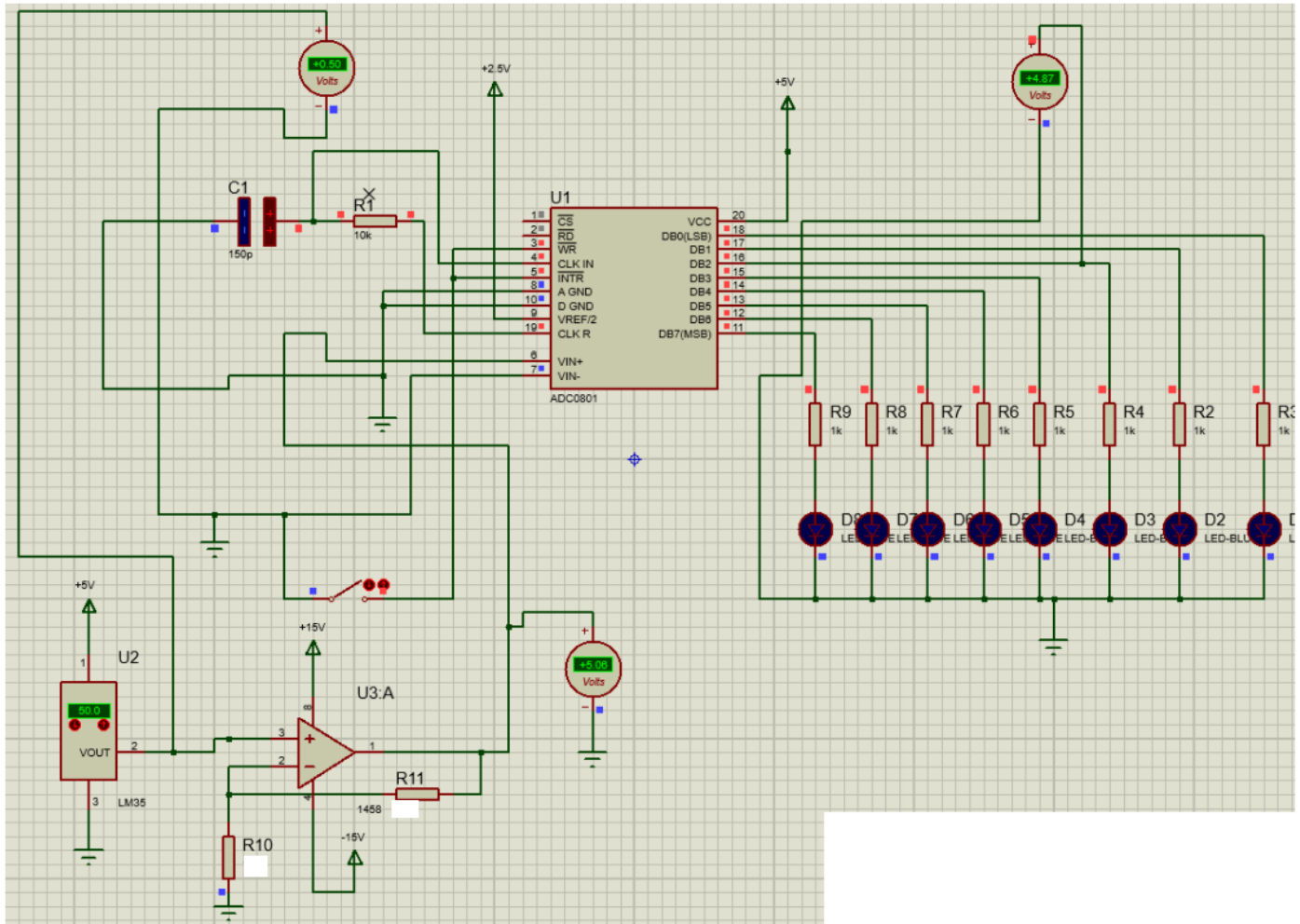
Exemple de montage :



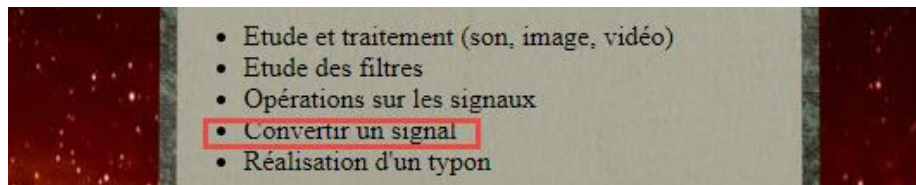
Nom :

Prénom :

- Maintenant, rajouter le convertisseur AN 8 bits afin d'obtenir le signal sur 8 bits (utiliser des leds pour contrôler la valeur de sortie sur 8 bits du CAN (0 ° C -> 0V -> 00000000 et 50 ° C -> 5V -> 11111111)

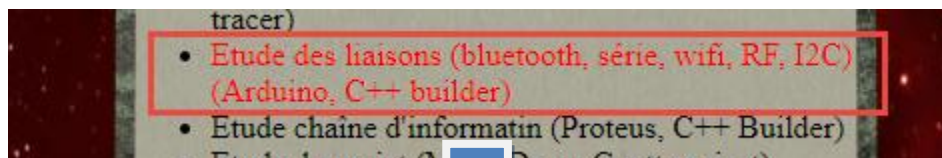


- TP d'aide



- Rajouter le module I2C et brancher la carte Arduino, ainsi que l'afficheur (Proteus)

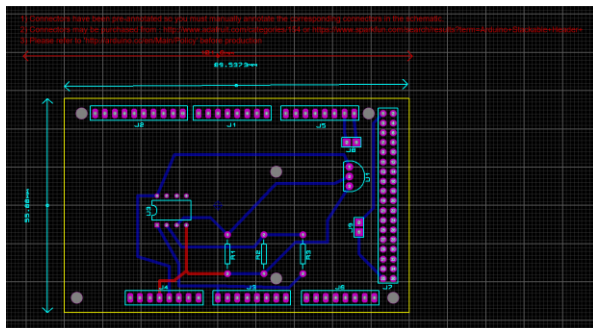
- TP d'aide



Nom :

Prénom :

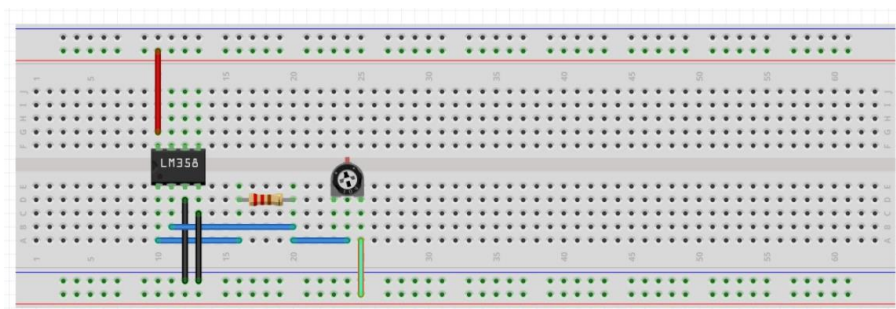
- Réaliser le programme afin d'afficher sur l'afficheur la température.
- Tester le programme sur Proteus et en réel. Faire des mesures avec un voltmètre et un thermomètre pour valider le montage
- Réaliser le PCB et imprimer le



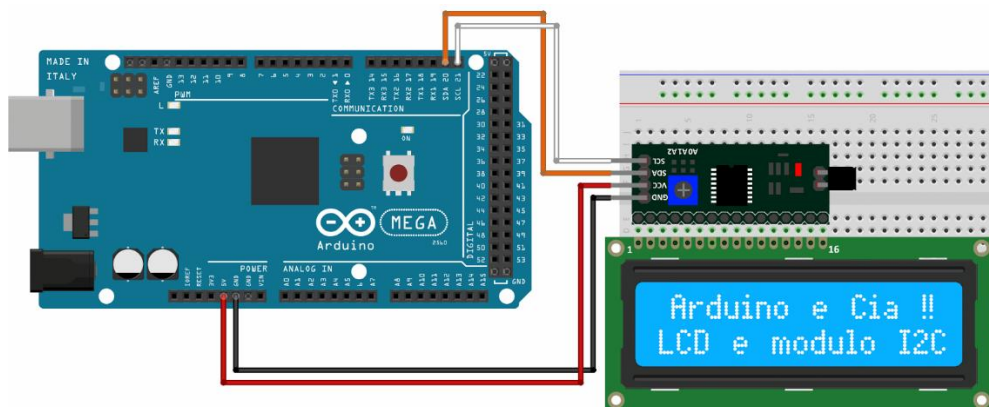
- Réaliser le schéma Sysml bloc interne
- Faire le branchement du montage sur Fritzing

Exemple Fritzing :

<http://fritzing.org/home/>

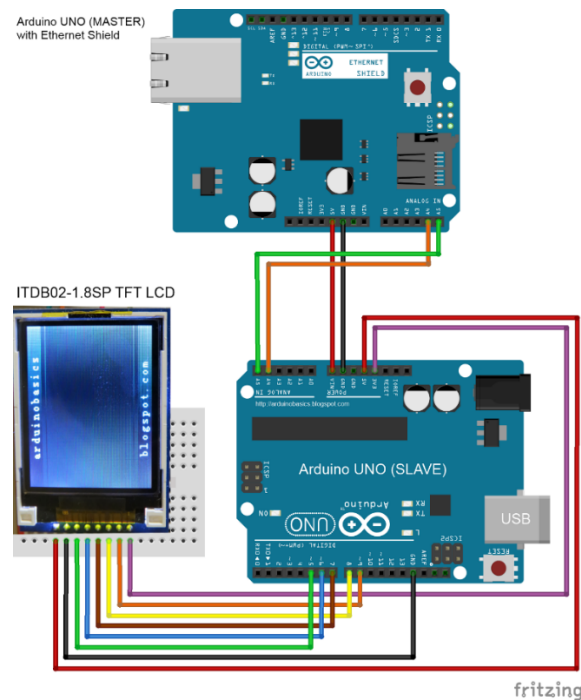


Montage afficheur LCD I2C



Nom :
Shield Internet

Prénom :



Attention d'utiliser des fils de couleur noire pour les masses

- Transmission de l'information dans la base de données (élève 2)
- Faire le branchement avec la carte Arduino muni du Shield internet
- Réaliser les programmes Arduino, Android et PHP pour respecter la demande du client
 - Fichier PHP :

- Un fichier pour récupérer la donnée envoyée par la carte Arduino en mode GET et l'enregistrer dans la base de données

Pour tester le fichier join.php, on pourra directement rentrer l'url suivant sur le navigateur :

<http://nom.alwaysdata.net/nomdossier/join.php?temperature=18> (adapter suivant votre programme)

- Un fichier (voir.php) pour afficher la liste des valeurs enregistrées, dans lequel est intégré un fichier PHP (graph.php) qui affiche le graphe.

```
<object type="text/html" width="600" height="600" data-dw-widget="Edge" data="graph.php">
</object></center>
```

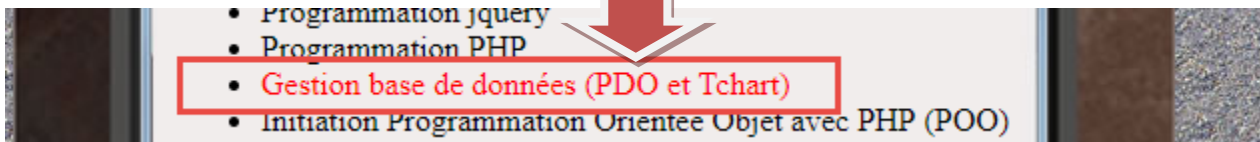
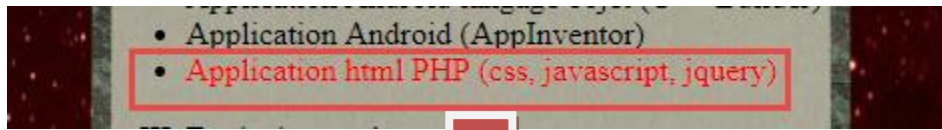
Programme pour intégrer le fichier graph.php dans le fichier voir.php. **Ne pas oublier d'intégrer la fonction de rafraichissement de voir.php pour mettre à jour automatiquement la page.**

```
<title>Regarder température</title>
<meta http-equiv="refresh" content="120">
<link href="voir.css" rel="stylesheet" type="text/css">
```

Nom :

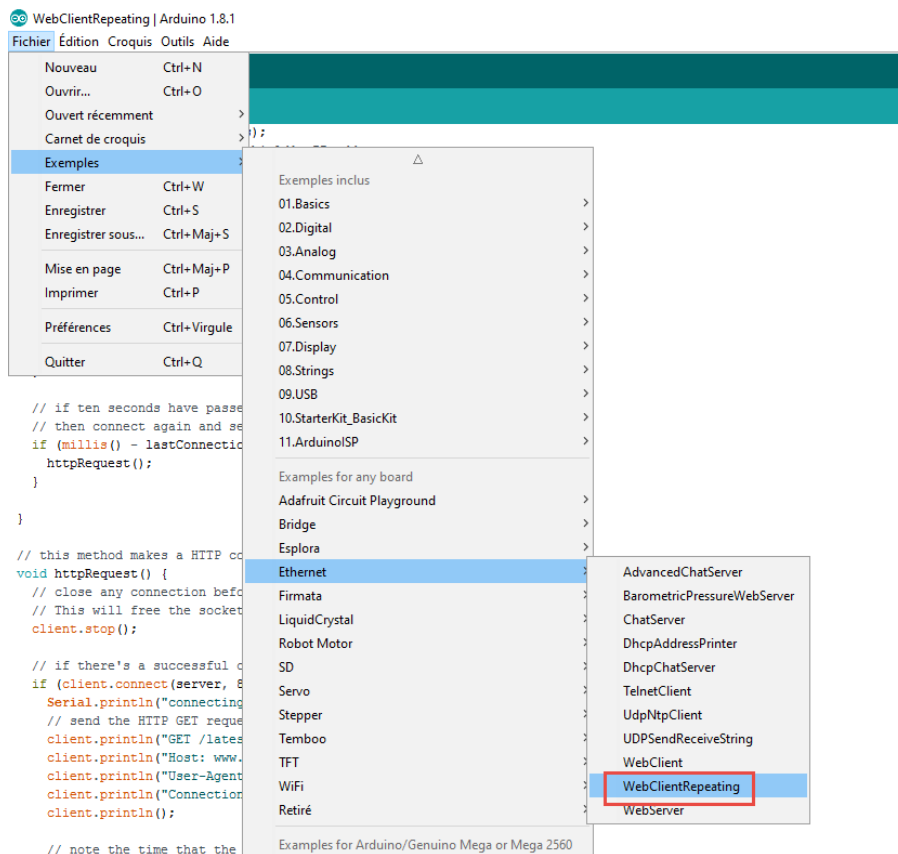
Prénom :

- TP d'aide



Remarques :

Pour la liaison internet on se basera sur l'exemple : Ethernet : WebClientRepeating sur Arduino



Pour envoyer la valeur, on pourra se baser sur les lignes de programme suivantes :

```
#include <SPI.h>
```

```
#include <Ethernet2.h>
```

```
// assign a MAC address for the ethernet controller.
```

```
// fill in your address here:
```

```
byte mac[] = {
```

```
    0x90, 0xA2, 0xDA, 0x10, 0xDE, 0xC1
```

Nom :
};

Prénom :

// fill in an available IP address on your network here,

// for manual configuration DNS:

IPAddress ip(192,168,333,61);

// fill in your Domain Name Server address here:

//IPAddress myDns(192, 168, 333, 61);

// initialize the library instance:

EthernetClient client;

char server[] = "nom.alwaysdata.net";

//IPAddress server(64,131,82,241);

unsigned long lastConnectionTime = 0; // last time you connected to the server, in milliseconds

const unsigned long postingInterval = 2000L; // delay between updates, in milliseconds

// the "L" is needed to use long type numbers

void setup() {

 // start serial port:

 Serial.begin(9600);

 while (!Serial) {

 ; // wait for serial port to connect. Needed for Leonardo only

 }

 // give the ethernet module time to boot up:

 delay(1000);

 // start the Ethernet connection using a fixed IP address and DNS server:

 if (Ethernet.begin(mac) == 0) {

 Serial.println("Failed to configure Ethernet using DHCP");

 // try to configure using IP address instead of DHCP:

 Ethernet.begin(mac, ip);

 }

 Serial.println("le client est sur l'adresse : ");

 Serial.println(Ethernet.localIP()); //on affiche l'adresse IP de la connexion

Nom :.....

Prénom :.....

// give the Ethernet shield a second to initialize:

delay(1000);

Serial.println("connecting...");

}

void loop() {

// if there's incoming data from the net connection.

// send it out the serial port. This is for debugging

// purposes only:

if (client.available()) {

char c = client.read();

Serial.write(c);

}

// if ten seconds have passed since your last connection,

// then connect again and send data:

if (millis() - lastConnectionTime > postingInterval) {

httpRequest();

}

}

// this method makes a HTTP connection to the server:

void httpRequest() {

// close any connection before send a new request.

// This will free the socket on the WiFi shield

client.stop();

// if there's a successful connection:

int value=18;

if (client.connect(server, 80)) {

Serial.println("connecting...");

// send the HTTP PUT request:

client.print("GET /dossier/fichier.php?temperature=");

client.print(value);

Nom :

Prénom :

```
// client.print("&humidite=");

// client.print("bonne");

client.println(" HTTP/1.1");

client.println("Host: nom.alwaysdata.net");


client.println("Connection: close");

client.println();


// note the time that the connection was made:

lastConnectionTime = millis();

}

else {

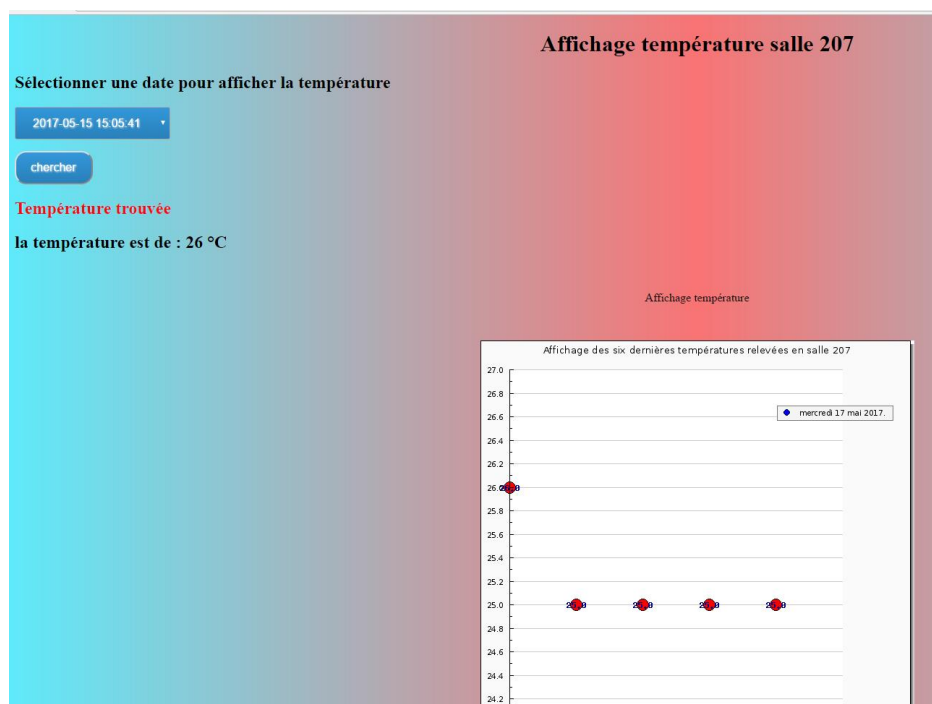
// if you couldn't make a connection:

Serial.println("connection failed");

}

}
```

Exemple page PHP :

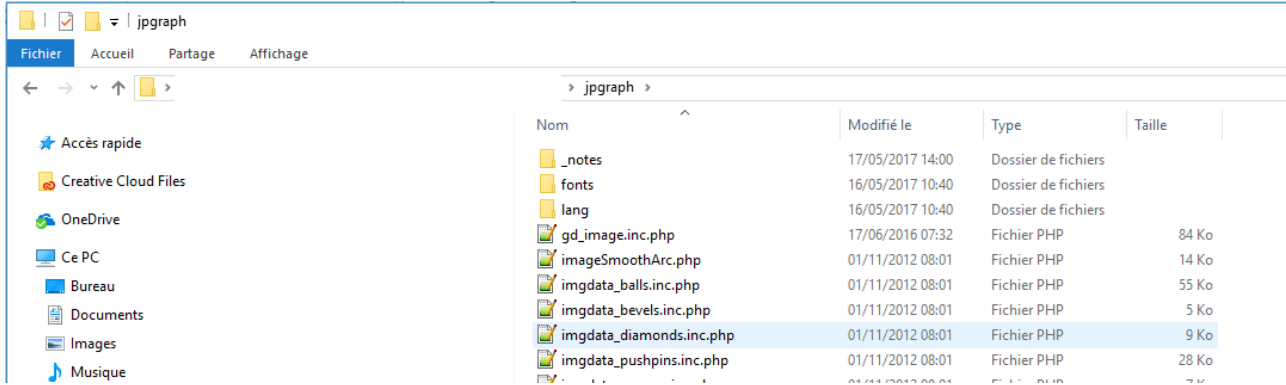


<http://www.sti2dsinhyrome.fr/qualiteairsalle/voir.php>

On pourra utiliser la bibliothèque jgraph pour les graphiques. Penser à créer un fichier « jgraph » où se trouve le fichier PHP. Dedans doivent se trouver les fichiers suivants :

Nom :

Prénom :



<http://jpgraph.net/download/>

Cette bibliothèque utilise des tableaux pour afficher les valeurs.

Etapas de travail :

1. Créer une boucle pour récupérer le nombre de valeur (x2) enregistrées dans la base.

```
$x2=0;
$req1=$pdo->query('SELECT * FROM qualite');
while($donne=$req1->fetch())
{
```

2. Connaissant le nombre de valeurs, récupérer les 6 dernières que vous allez mettre dans deux tableaux pour x et y

Pour cela il faudra d'abord définir deux tableaux pour x(date), y(température). Puis récupérer les valeurs dans la base de données et les enregistrer dans les deux tableaux avec une boucle « while ».

```
$date1=array();
$temperature=array();

$req=$pdo->prepare('SELECT * FROM qualite LIMIT :v1,6');
$req->bindValue(':v1', intval($x2), PDO::PARAM_INT);
$req->execute();
while($donne=$req->fetch())
{
    $temperature[]=$donne['temperature'];
```

On utilisera différentes couleurs pour les cercles en fonction de la température.

- Bleue : < 17 °C
- Verte : 17 < t < 24 °C
- Rouge : > 24 °C

Pour afficher le graphique, vous devez inclure dans le fichier PHP les fichiers suivants :

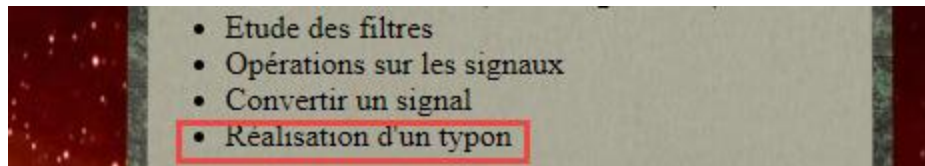
Nom :

Prénom :

```
<?php // content="text/plain; charset=utf-8"
// $Id: balloonex1.php,v 1.5 2002/12/15 16:08:51 aditus Exp $

require_once ('jpgraph/jpgraph.php');
require_once ('jpgraph/jpgraph_scatter.php');
require_once ('jpgraph/jpgraph_line.php');
require_once ('jpgraph/jpgraph_date.php');
// Some data
$host="";
$dbname="";
$login="";
$password="";
try
{
    $pdo=new PDO('mysql:host='.$host.';dbname='.$dbname.'', $login,$password );
}
catch (Exception $erreur)
{
    die('Erreur:'.$erreur->getMessage());
}
$x2=0;
$req1=$pdo->query('SELECT * FROM qualite');
```

- Réaliser le boîtier dans lequel sera installé le PCB et imprimer une image du boîtier
 - TP d'aide



- Réalisation d'une application Windows (élève 3)
 - Réaliser une application Windows qui va permettre de contrôler le fonctionnement du système. Vous utiliserez la liaison série de la carte Arduino.
 - TP d'aide
- <http://www.sti2dsinhyrome.fr/doc%20cours/liaison/serie/docserie.html>
- Réaliser un dossier technique présentant votre travail

Tp à réaliser pour la compréhension du projet (obligatoire):

- Application Html, Css, Javascript, JQuery
 - <http://www.sti2dsinhyrome.fr/doc%20cours/html/htmlcss/dochtmlcss.html>
 - <http://www.sti2dsinhyrome.fr/doc%20cours/html/javascript/docjavascript.html>
 - <http://www.sti2dsinhyrome.fr/doc%20cours/html/php/docphp.html>
 - <http://www.sti2dsinhyrome.fr/doc%20cours/html/pdo/docpdo.html>
- Etude d'un réseau
 - <http://www.sti2dsinhyrome.fr/doc%20cours/reseau/reseau/tpreseau.html>
- Application Arduino
 - <http://www.sti2dsinhyrome.fr/docmicrocontroleur.html>
- Etude chaîne d'information :

Nom :

Prénom :

- Les capteurs :

<http://www.sti2dsinhyome.fr/doc%20cours/capteur/capteurdrone/doccapteurdrone.html>

- Les filtres :

<http://www.sti2dsinhyome.fr/doc%20cours/filtre/filtre/doctpfiltre.html>

- Opérations sur les signaux

<http://www.sti2dsinhyome.fr/doc%20cours/op%C3%A9ration/op%C3%A9ration/docop%C3%A9ration.html>

- Convertir les signaux

<http://www.sti2dsinhyome.fr/doc%20cours/convertisseur/convertisseur/docconvertisseur.html>

- Les liaisons avec fil

- Liaison I2C

<http://www.sti2dsinhyome.fr/doc%20cours/liaison/i2c/i2c.html>

- Liaison série (pour 3 élèves)

<http://www.sti2dsinhyome.fr/doc%20cours/liaison/serie/docserie.html>