**Projet Arduino chauffage**

Description

L’installation permet de chauffer une résistance de 100Ω pendant une durée déterminée.

Le but est de réguler la température de façon proportionnelle (PID).

Le programme doit fonctionner selon le concept du graph d’état.

Informations complémentaires

Tout le code ainsi que des informations supplémentaires du projet sont disponible sur GitHub : <https://github.com/CrBast/distillation_column>

Composants

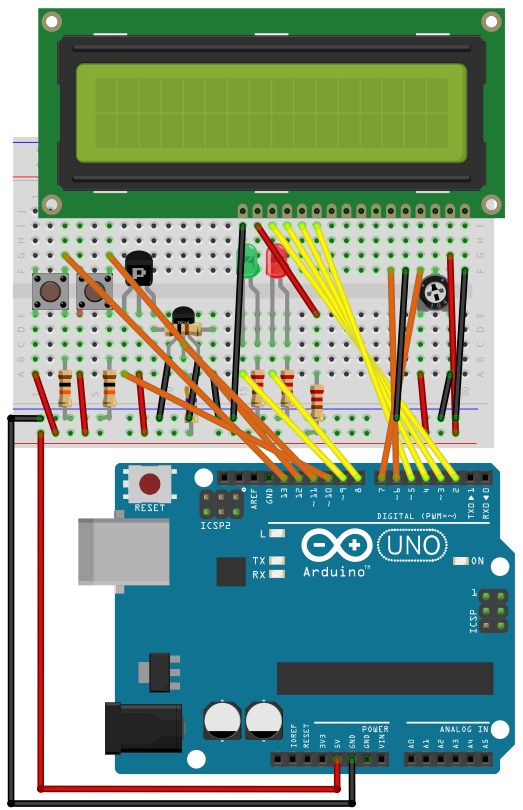
* Ecran LCD
* Capteur de température (Dallas – DS18B20)
* Transistor (BD675)
* Résistance de 100 Ω

Dépendances

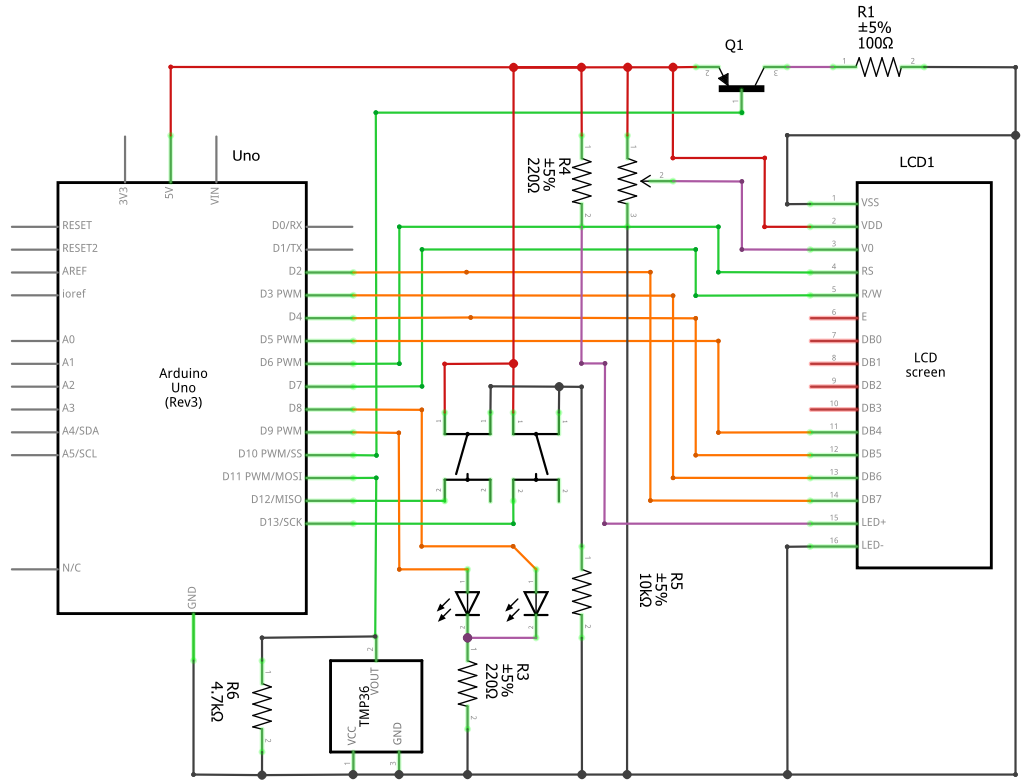
Voici les librairies open source utilisées :

* LiquidCrystal
* OneWire
* DallasTemperature

**Partie Arduino**



Vue schématique



**DS18B20**

Vue plaquette d’expérimentation

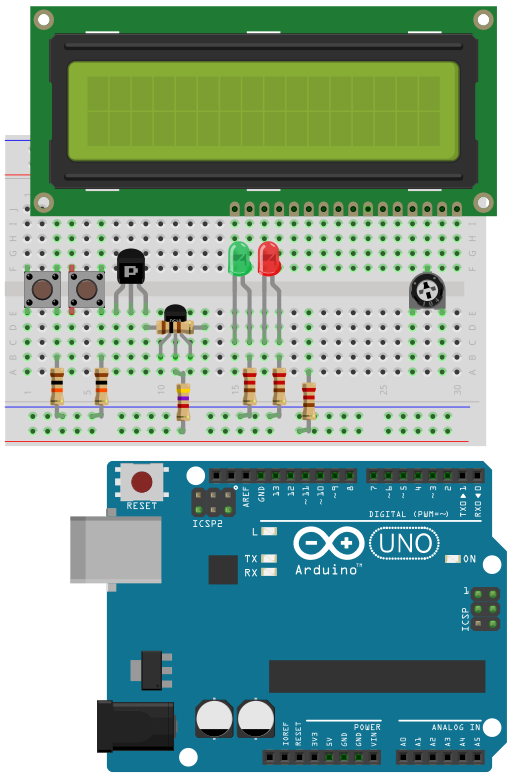
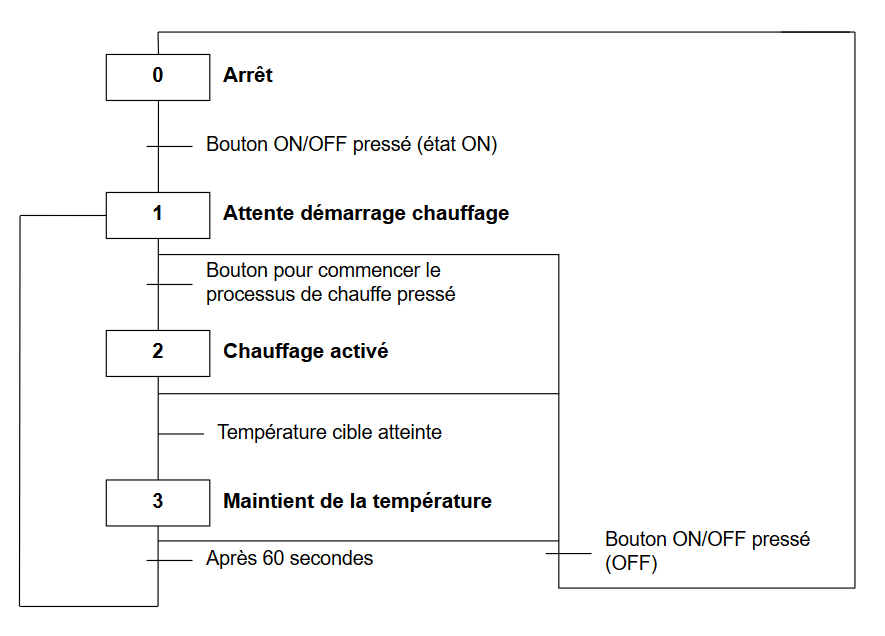
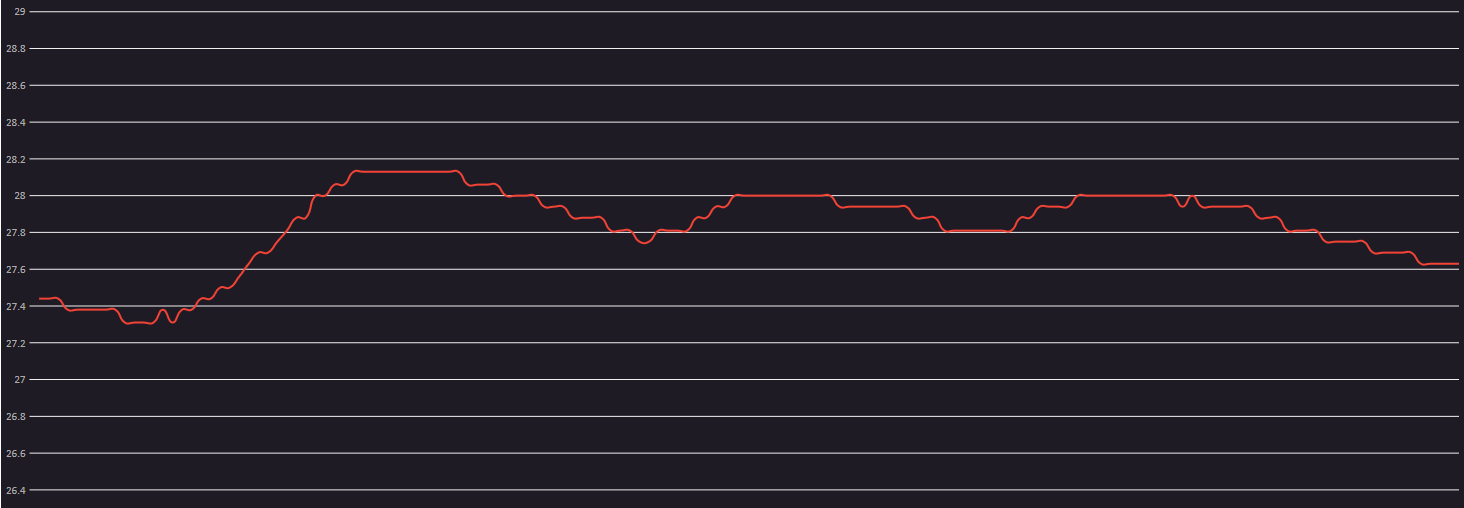


Diagramme états-transitions



Graphique test température (PID)



Etat 1

Etat 3

Etat 2

Etat 1

Code

Github : <https://github.com/CrBast/distillation_column/blob/master/arduino/arduino.ino>

**Application Windows**

Description

L’application permet de visualiser en temp-réel diverses informations concernant le montage Arduino.

Il est possible d’exporter les données de température en CSV. Celles-ci se trouvent dans le répertoire /data/ à la racine de l’application.

L’Application a été développée en WPF (C#) et est seulement disponible pour Windows(32/64bits).

Le code complet de l’application se trouve sur GitHub : <https://github.com/CrBast/distillation_column/tree/master/windows-manager>

Convention de transfert de données Arduino – PC : <https://github.com/CrBast/distillation_column/wiki>

Dépendances

Voici les librairies open source utilisées

* LiveCharts (<https://lvcharts.net>)

Interface graphique

Les informations disponibles en temps réel :

* Température du capteur
* Etat (Selon diagramme états-transitions)
* Puissance de chauffe
* Toutes les informations transmises par l’Arduino

