

Tutoduino.fr

Tutoriels pour Arduino et Raspberry



Régulation de chauffage par PID avec un Arduino

★★★★★ 4.6 (37)

Dans ce tuto nous allons voir comment réguler un petit système de chauffage par PID avec un Arduino. Nous utiliserons une plaque chauffante PTC en aluminium que l'on va contrôler avec un relai statique SSR. Une sonde de température DS18B20 sera collée sur la plaque chauffante et permettra de mesurer sa température.

Liste du matériel utilisé

- [Arduino Uno](#)
- [Plaque chauffante PTC 24V 70°C](#)
- [Capteur de température DS18B20 3 fils](#)
- [Relai statique DC-DC](#)



Matériel utilisé pour ce tutoriel

C'est quoi une régulation PID ?

Un [régulateur PID](#) est un système de contrôle en boucle fermée. **PID** est l'abréviation de **P**roportionnelle **I**ntégrale et **D**érivée, les trois actions effectuées par ce régulateur. Le PID peut très bien s'expliquer de façon intuitive. Le but de [cet article](#) est d'expliquer comment marche une régulation PID sans entrer dans les calculs ni sans utiliser les fonctions de transfert du système et autres équations.

Nous utiliserons la librairie **PID** disponible dans l'IDE Arduino. La version courante lors de la rédaction de ce tuto est le 1.2.0.



Librairie PID utilisée dans ce tutoriel

Notre tuto est basé sur l'exemple PID_RelayOutput fourni avec la librairie dont voici le code :

```

/*****
 * PID RelayOutput Example
 * Same as basic example, except that this time, the output
 * is going to a digital pin which (we presume) is controlling
 * a relay. the pid is designed to Output an analog value,
 * but the relay can only be On/Off.
 *
 * to connect them together we use "time proportioning"
 *****/

```

```

* control" it's essentially a really slow version of PWM.
* first we decide on a window size (5000mS say.) we then
* set the pid to adjust its output between 0 and that window
* size. lastly, we add some logic that translates the PID
* output into "Relay On Time" with the remainder of the
* window being "Relay Off Time"
*****/
#include <PID_v1.h>
#define PIN_INPUT 0
#define RELAY_PIN 6
//Define Variables we'll be connecting to
double Setpoint, Input, Output;
//Specify the links and initial tuning parameters
double Kp=2, Ki=5, Kd=1;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
int WindowSize = 5000;
unsigned long windowStartTime;
void setup()
{
  windowStartTime = millis();
  //initialize the variables we're linked to
  Setpoint = 100;
  //tell the PID to range between 0 and the full window size
  myPID.SetOutputLimits(0, WindowSize);
  //turn the PID on
  myPID.SetMode(AUTOMATIC);
}
void loop()
{
  Input = analogRead(PIN_INPUT);
  myPID.Compute();
  /*****
  * turn the output pin on/off based on pid output
  *****/
  if (millis() - windowStartTime > WindowSize)
  { //time to shift the Relay Window
    windowStartTime += WindowSize;
  }
  if (Output < millis() - windowStartTime) digitalWrite(RELAY_
else digitalWrite(RELAY_PIN, LOW);
}

```

Attention toutefois, ce code fonctionne avec un relais NC (Normally Closed). Avec un relais NC la plaque chauffante est alimentée lorsque le relais est fermé et donc que lorsque la broche de sortie est à l'état bas (LOW). Dans mon tutoriel, le relai que j'utilise est un relai NO (Normally Opened). La broche de sortie doit donc être à l'état haut (HIGH) pour que la plaque chauffante soit alimentée.

Le bout de code à la fin du programme devra donc être modifié de la sorte :

```
if (Output < millis() - windowStartTime)
    digitalWrite(RELAY_PIN, LOW);
else
    digitalWrite(RELAY_PIN, HIGH);
```

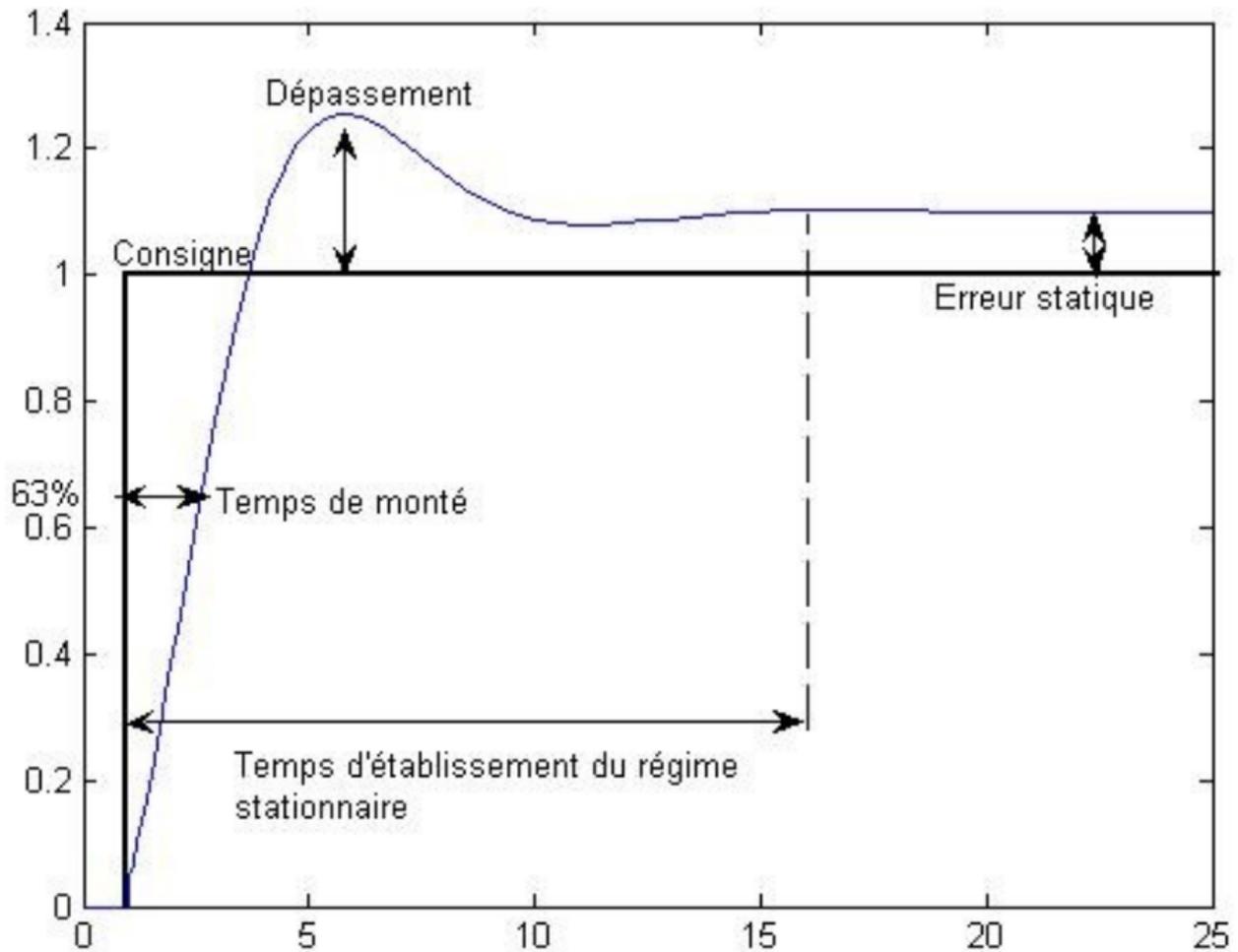
?

Réglage des paramètres de la régulation PID

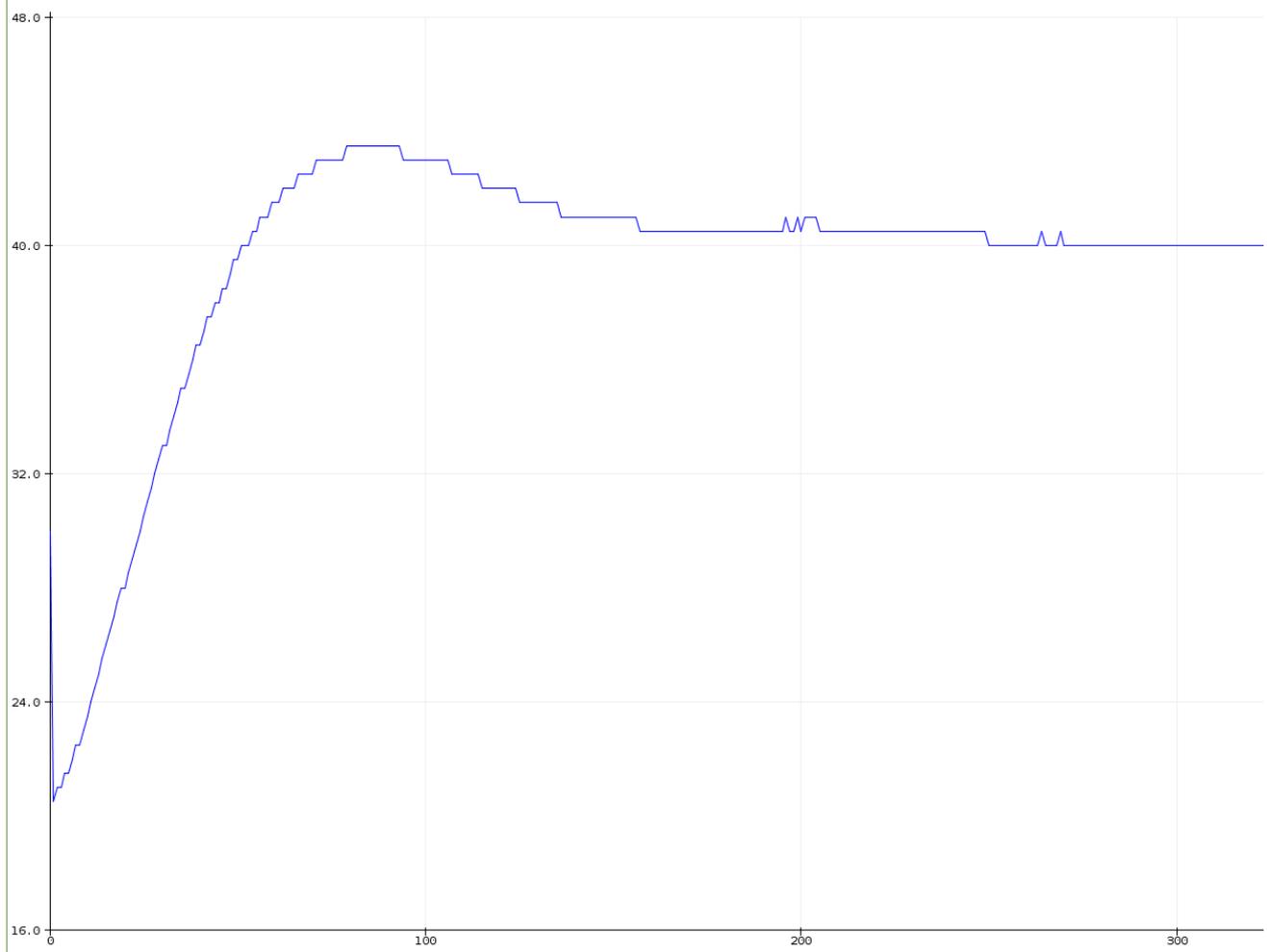
La régulation PID doit être paramétrée, elle dépend en effet du système qui est régulé. Il va donc falloir déterminer la valeur des coefficients **Kp, Ki et Kd** de notre régulateur.

Comme expliqué dans l'article mentionné plus haut le réglage peut s'effectuer en 2 étapes :

- Il faut tout d'abord mettre en place un simple régulateur proportionnel (les coefficients Ki et Kd sont donc nuls). Par essais/erreurs, il faut régler le coefficient **Kp** afin d'améliorer le temps de réponse du système. C'est-à-dire qu'il faut trouver un Kp qui permette au système de se rapprocher très vite de la consigne tout en faisant attention de garder la stabilité du système : il ne faut pas que le système réponde très vite tout en oscillant beaucoup !
- Une fois ce coefficient réglé, on peut passer au coefficient **Ki**. Celui-là va permettre d'annuler l'erreur statique finale du système afin que celui-ci respecte exactement la consigne. Il faut donc régler Ki pour avoir une réponse exacte en peu de temps tout en essayant de minimiser les oscillations apportées par l'intégrateur !
- Enfin, si le système n'est pas stable, on peut passer au dernier coefficient **Kd** qui permet de rendre le système plus stable et permet de diminuer les oscillations.

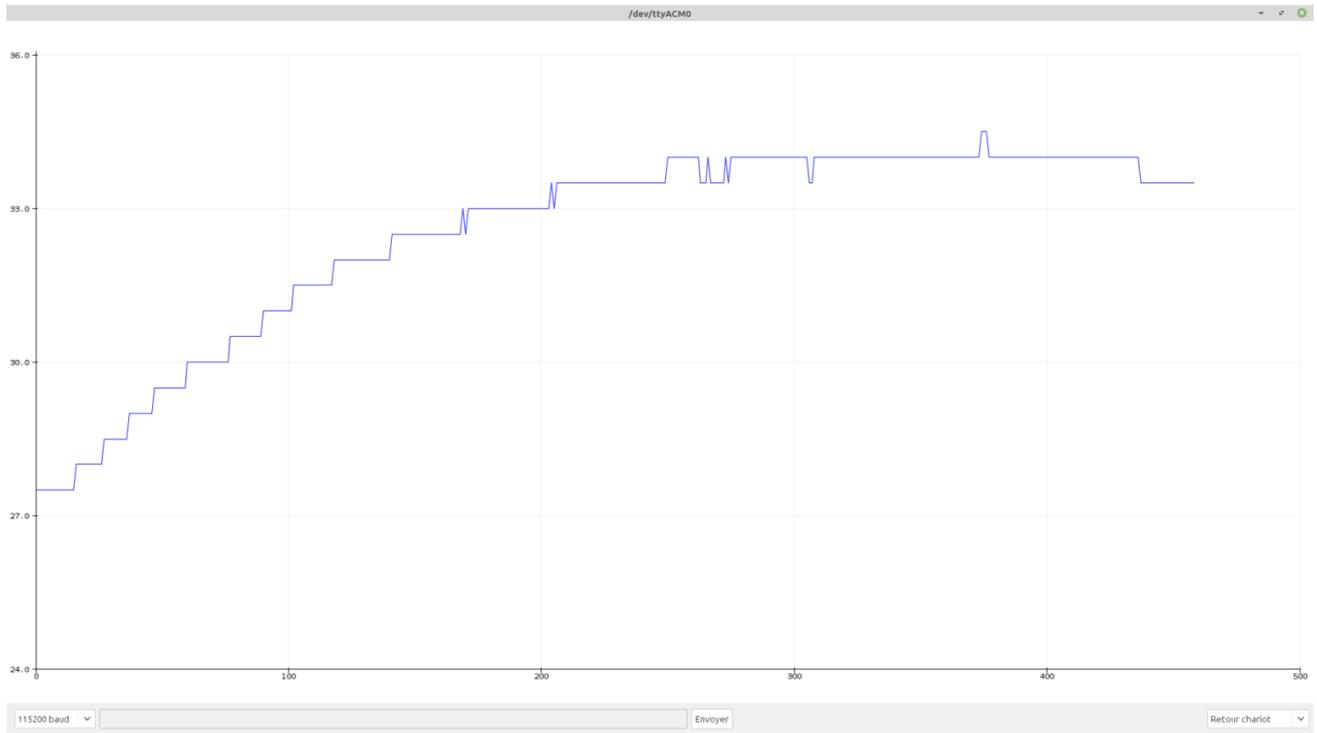


Après plusieurs essais, les paramètres $K_p=65$, $K_i=0.71$ et $K_d=0$ semblent plutôt corrects. La montée est assez rapide, le dépassement acceptable (3°C au dessus de la consigne) et le régime stationnaire est atteint rapidement avec une température stable de 40° au bout de 3 minutes environ.

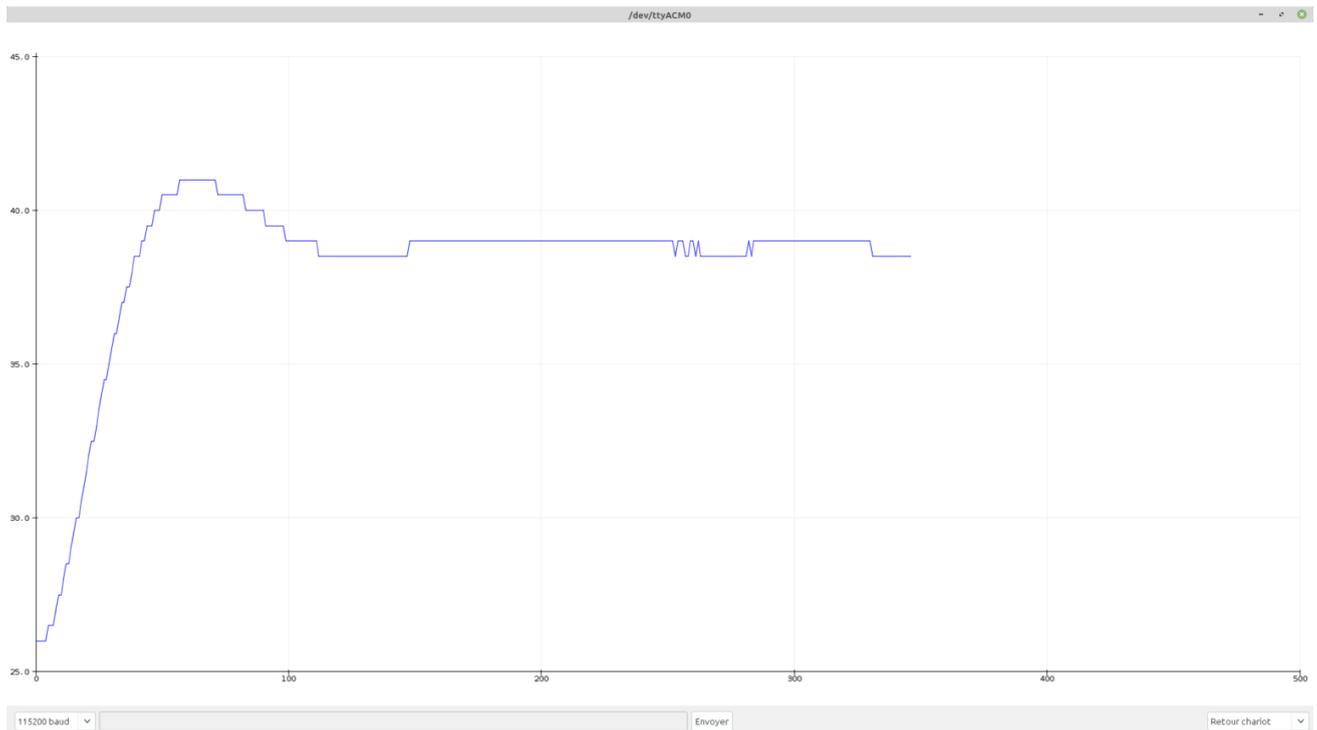


Régulation PID avec une consigne de 40°C et les paramètres $K_p=65$, $K_i=0.71$, $K_d=0$

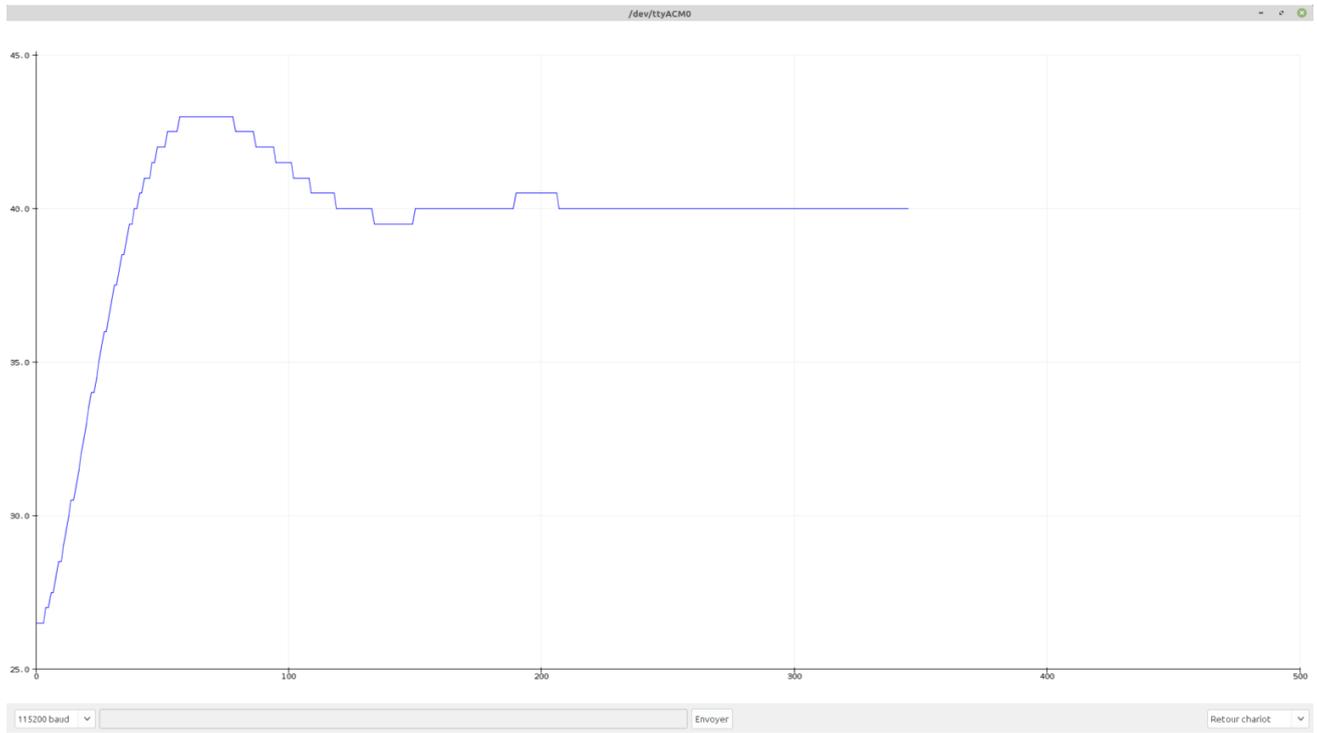
Voici quelques exemples de réglages PID qui ne conviennent pas aux caractéristiques du système.



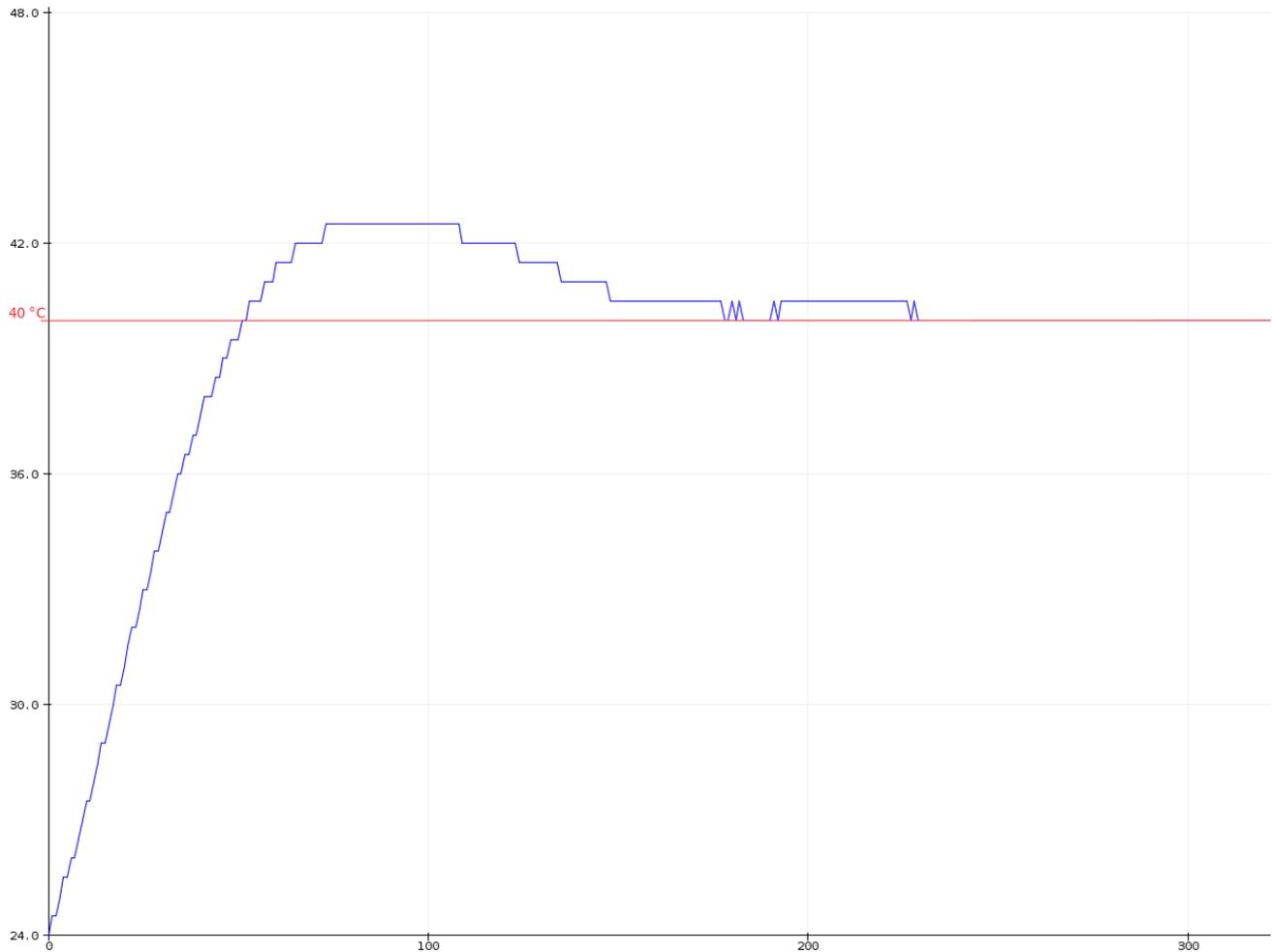
Avec les paramètres $K_p=10$, $K_i=0$, $K_d=0$ le système n'arrive pas à atteindre la consigne de 40°C ; le paramètre K_p est trop faible



Avec les paramètres $K_p=100$, $K_i=0$, $K_d=0$ le système se stabilise à 39°C ; le paramètre K_i doit être augmenté pour compenser l'erreur statique



Avec les paramètres $K_p=100$, $K_i=1$, $K_d=0$ le système se stabilise à 40°C au bout d'environ 210 secondes et le dépassement (3°C de plus que la consigne) reste acceptable pour mon système



Autre réglage de paramètres avec $K_p=55$, $K_i=0.8$, $K_d=0.3$

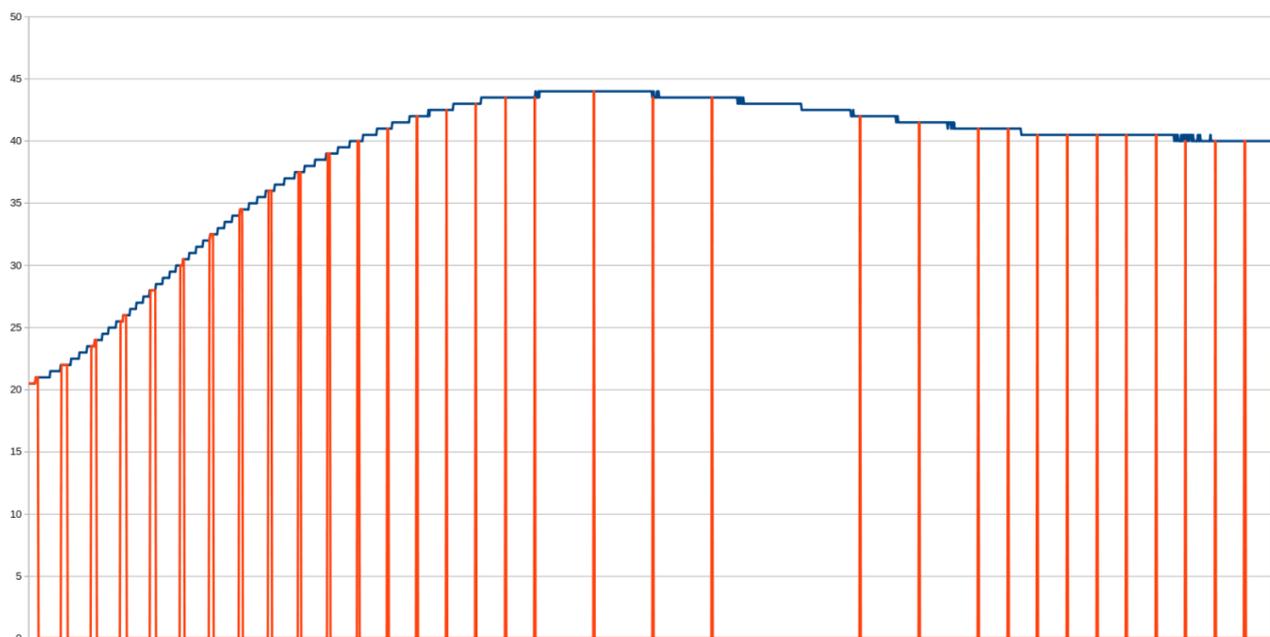
Visualisation de la consigne de sortie de la régulation PID

Dans le graphique suivant, vous pouvez visualiser deux courbes :

- La courbe bleue représente l'entrée de la boucle de régulation PID. Elle correspond à la valeur retournée par le capteur de température.
- La courbe rouge représente l'état du relais. Lorsque cette valeur est à 0, cela signifie que le relais est ouvert (la plaque ne chauffe pas). Lorsque cette valeur est différente de 0, cela signifie que le relais est fermé (la plaque chauffe).

On voit que les impulsions sont plus rapprochées et plus longues dans la phase de montée en température. Après une pause qui permet d'atteindre la température de

consigne, ces impulsions deviennent brèves et leur fréquence permet de maintenir le système à une température constante.



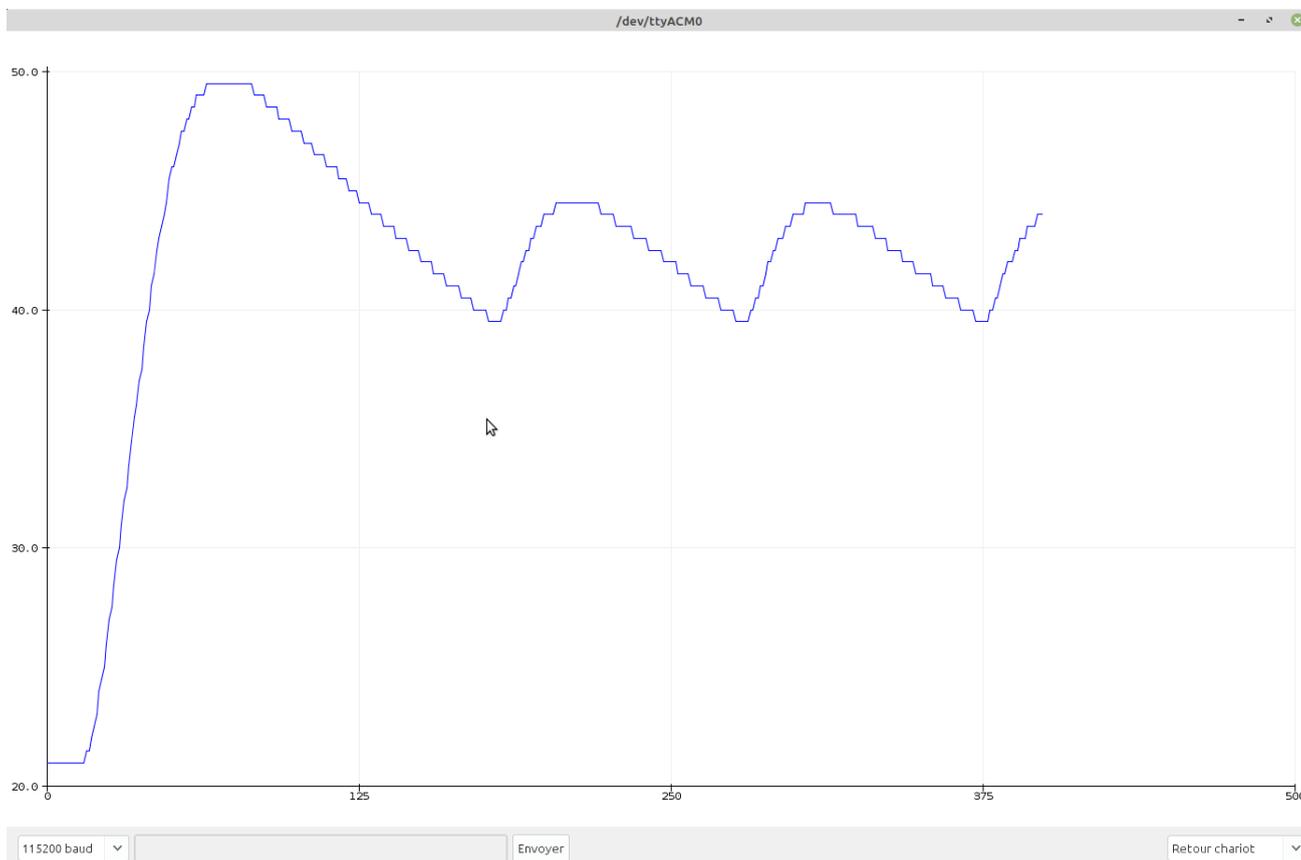
A t-on vraiment besoin d'une régulation PID ?

Vous vous posez sûrement la question de savoir si une simple régulation par seuil pourrait convenir. En allumant la plaque chauffante dès que la température est inférieure à 40°C et l'éteignant dès que la température est supérieure ou égale à 40°C. Comme ce simple code par exemple :

```
if (temp >= 40) {  
    commande_relai(LOW);  
} else {  
    commande_relai(HIGH);  
}
```

?

Et bien le résultat est clairement visible sur la courbe suivante. Le dépassement est élevé, la température monte à 50°C soit 10 °C de plus que la consigne de 40°C. La régulation ne se stabilise pas, elle oscille continuellement entre 39°C et 45°C...

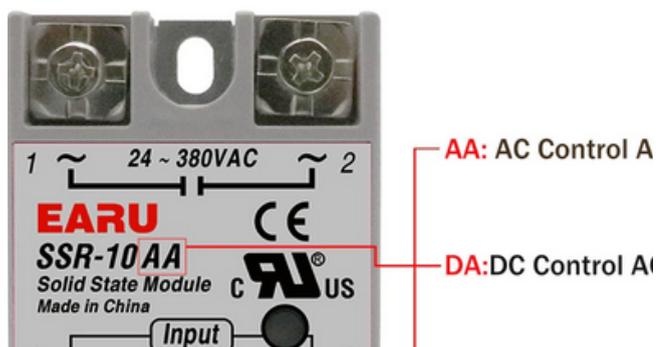


Exemple de régulation basique avec un déclenchement sur seuil

Relai statique SSR

Un relai statique SSR nécessite peu de courant pour être contrôlé (10mA à 5V pour le mien). Il peut donc être alimenté directement par une broche de l'Arduino. Je vous invite à lire [cet article](#) qui explique le contrôle par relais.

Le relais que j'utilise permet de contrôler un courant de 10 A sous une tension continue comprise entre 12 V et 220 V. La tension de contrôle est également continue et comprise entre 3 V et 32 V. Il s'agit de la référence SSR-10 DD.



Signification des 2 lettres sur la référence des relais

SSR



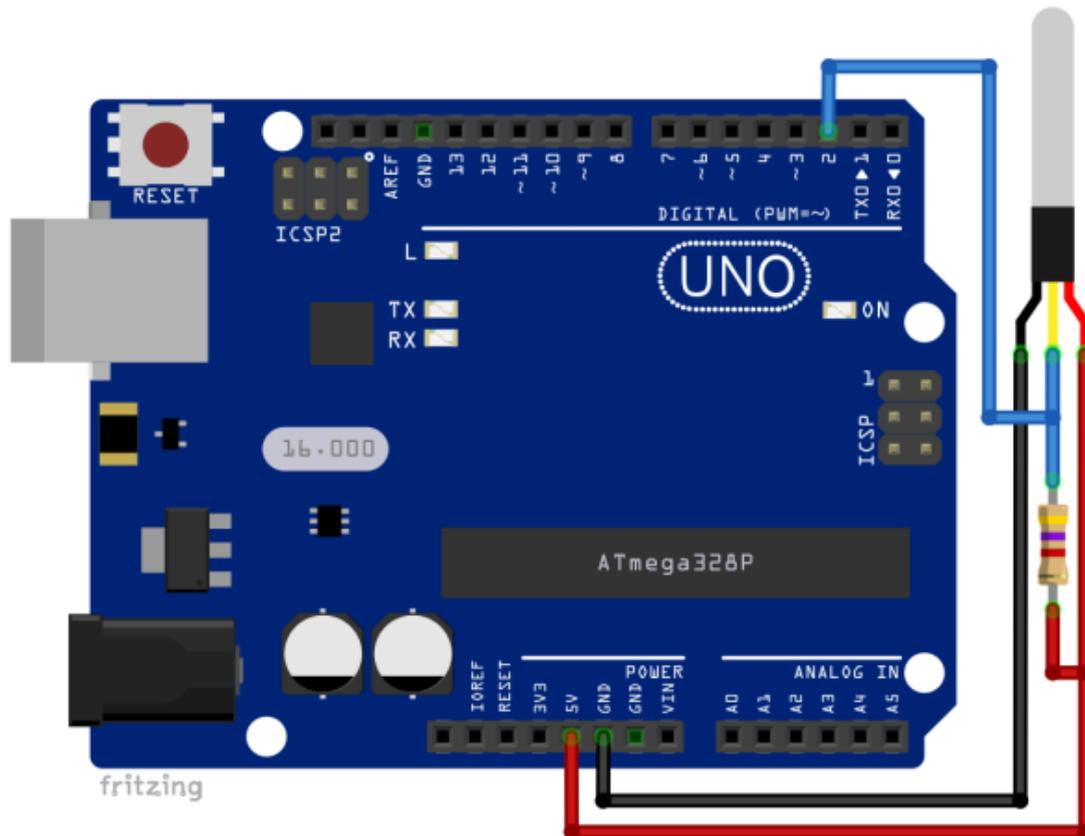
Relais SSR-10 DD utilisé dans ce tutoriel



Le capteur de température DS18B20

Ce capteur de température permet de mesurer des températures comprises entre -55 °C et $+125\text{ °C}$. Il est équipé d'un boîtier en acier inoxydable que l'on peut facilement coller sur la plaque chauffante grâce à de la [colle thermique](#).

Ne pas oublier de rajouter une [résistance de rappel](#) "pull-up" de $4,7\text{ k}\Omega$ comme indiqué dans la documentation du capteur.



Pin DS18B20 > Pin Uno

DATA	>	D2	Câble Bleu
VCC	>	5V	Câble Rouge
GND	>	GND	Câble Noir

NOTE : Une résistance de $4,7k\Omega$ est connectée entre la broche *DATA* et *VCC*.

Schéma de branchement du capteur DS18B20 sur l'Arduino Uno

J'ai utilisé la librairie **DallasTemperature** pour ce tuto.



Librairie DallasTemperature utilisée dans ce tutoriel

Cette librairie inclus de nombreux programmes d'exemple pour ce capteur. Le tuto se base sur le programme Simple dont voici le code :

```
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>
// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 2
// Setup a oneWire instance to communicate with any OneWire de
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
/*
 * The setup function. We only start the sensors here
 */
void setup(void)
{
  // start serial port
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo")
  // Start up the library
  sensors.begin();
}
/*
 * Main function, get and show the temperature
 */
void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temp
  // request to all devices on the bus
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get te
  Serial.println("DONE");
  // After we got the temperatures, we can print them here.
  // We use the function ByIndex, and as an example get the te
  float tempC = sensors.getTempCByIndex(0);
  // Check if reading was successful
  if(tempC != DEVICE_DISCONNECTED_C)
  {
    Serial.print("Temperature for the device 1 (index 0) is: "
    Serial.println(tempC);
  }
  else
  {
    Serial.println("Error: Could not read temperature data");
  }
}
```

Le programme complet

Voici le programme utilisé pour la régulation. Il utilise la librairie PID pour la régulation ainsi que la librairie DallasTemperature pour la mesure du capteur de température.

```
/*
  PID regulation
  Created March 2022
  by Tutoduino
*/
// For PID regulation
#include <PID_v1.h>
// For DS18B20 sensor
#include <OneWire.h>
#include <DallasTemperature.h>
// DS18B20 sensor is on PIN 2 of the Arduino
#define ONE_WIRE_BUS 2
// Solid state relay is on PIN 4 of the Arduino
#define RELAY_PIN 4
// Sensor objects
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
// PID variables
double Setpoint, Input, Output;
int WindowSize = 5000;
unsigned long windowStartTime, lastPrintTime;
// PID tuning parameters
double Kp=65, Ki=0.71, Kd=0;
// PID object
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
// Function that controls activation or deactivation
// of the relay. The LED built into the Arduino is on when
// the relay is active, otherwise it is off.
void relay_state(bool state) {
    digitalWrite(RELAY_PIN, state);
    digitalWrite(LED_BUILTIN, state);
}
// Function that print temperature each second
void printTemperatureEverySec(double temp) {
    if ((millis()-lastPrintTime) > 1000) {
        lastPrintTime = millis();
        Serial.println(temp);
    }
}
// Setup function that runs once at startup
void setup()
{
    // Define relay and led pins as output
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(LED_BUILTIN, OUTPUT);

    // Set the relay to low state
    relay_state(LOW);
}
```

```
Serial.begin(115200);
// Store current time in variables used by PID
// and display functions
windowStartTime = millis();
lastPrintTime = millis();

// Setpoint for the PID temperature control 40 degrees Celci
Setpoint = 40;
// Tell the PID to range between 0 and the full window size
myPID.SetOutputLimits(0, WindowSize);
// Turn the PID on
myPID.SetMode(AUTOMATIC);
}
// Main loop function
void loop()
{
  // Get the temperature from the sensor
  sensors.requestTemperatures();
  Input = sensors.getTempCByIndex(0);
  printTemperatureEverySec(Input);
  // Compute the PID
  myPID.Compute();
  /******
   * turn the output pin on/off based on pid output
   * *****/
  if (millis() - windowStartTime > WindowSize)
  {
    //time to shift the Relay Window
    windowStartTime += WindowSize;
  }
  if (Output < millis() - windowStartTime) {
    relay_state(LOW);
  }
  else
    relay_state(HIGH);
}
```

Votre avis compte !



Note moyenne : 4.6 / 5. Nombre de votes : 37

(29 commentaires)



Alain AUCH dit :

27 juin 2024 à 7h41

Merci pour ce tutoriel.

Technicien électronicien de 1964 à 1992, j'ai assuré (et survécu) à de nombreux virages technologiques depuis les tubes à vide jusqu'aux microprocesseurs, mais j'ai principalement travaillé sur des circuits analogiques. Je n'ai jamais appris à programmer en C ou autre langage.

A 80 ans, passionné par la sculpture en céramique, j'ai construit un four et je me suis mis en tête de mesurer la température l'aide d'une carte Arduino Uno Rev. 3, d'un circuit Adafruit MAX 31856 et d'un thermocouple type S.

Il y a 2 paramètres qui m'intéressent : la température en °C et la pente de montée en température en °C / heure.

Dans un premier temps je peux utiliser le moniteur de IDE pour afficher les résultats.

Dans un deuxième temps rajouter un afficheur LCD connect à la carte pour cela.

Pouvez-vous me suggérer un tutoriel qui puisse m'aider ?

Merci d'avance.

[Répondre](#)



Tutoduino dit :

2 juillet 2024 à 19h22

Ce tuto pourrait vous aider dans un premier temps pour tracer la courbe via le moniteur IDE <https://tutoduino.fr/debuter/mesurer-et-tracer-une-tension/>

[Répondre](#)



AANOMADE dit :

5 juillet 2024 à 18h30

Merci pour votre réponse.

J'ai l'impression d'avoir mis mon post de présentation au mauvais endroit dans le forum où le bar.

Sur la page d'accueil ça ne paraît pas très clair.

[Répondre](#)



FlorianB dit :

22 janvier 2024 à 15h45

Bonjour,

Merci pour ce tuto.

Je me posais la question d'utiliser un mosfet au lieu du relais pour des questions de coûts. Est-ce que selon vous cela poserait un problème ? Ca nécessite une PCB adaptée certes mais en terme de place et de coûts c'est incomparable.

[Répondre](#)



Tutoduino dit :

27 janvier 2024 à 6h14

Bonjour.

Oui bien entendu, moins cher et plus petit. C'est d'ailleurs ce que j'ai fait ensuite mais je n'ai pas encore eu le temps de mettre à jour mon tuto.

Voir la fin de ce tuto si vous le souhaitez :

<https://tutoduino.fr/tutoriels/controler-ventilateur-brushless-arduino/>

[Répondre](#)



Sahli dit :

3 décembre 2023 à 17h52

bonsoir,

merci pour ce travail très intéressant que je voudrai adapter pour un TP avec mes élèves de 2ème année post bac. je voulais savoir si je pouvais avoir une photo du circuit complet ou du schémas de montage. mon idée est de rajouter 3 potentiomètres un pour Kp, un pour Ki et un pour Kd ainsi en jouant sur celui de Kp par exemple, ils verront l'effet sur la température de la plaque et ça devient plus parlant expérimentalement.
d'avance merci

[Répondre](#)



YASSIN dit :

2 juin 2023 à 23h42

bonjour

je travail pour regulation pid avec arduino

esp8266 +relais statique +capteur de temperature LM35 +element chauffants

probleme element nest pas chauffer car relais ne passe pas la tension

le code nest fonction pas ??

[Répondre](#)



Tutoduino dit :

3 juin 2023 à 15h27

Bonjour.

La tension de sortie de l'ESP8266 n'est pas suffisante pour faire basculer le relais

statique probablement. Je vous invite à regarder au bas de cette page [https://
tutoduino.fr/debuter/controler-un-relai/](https://tutoduino.fr/debuter/controler-un-relai/), j'indique comment résoudre ce problème.

J'ai testé et cela fonctionne bien avec un ESP8266 et un relais statique SSR-10 DD.

[Répondre](#)



ness dit :

6 mai 2023 à 13h40

bonjour ,

tous d'abord merci pour votre tuto, cependant puisque je suis encore qu'une débutante

je voulais savoir comment faire quand on a une résistance chauffante de 650W que je

dois contrôler avec un relais statique ssr de 25A sachant que la température est mesurer

par un thermocouple type K. le problème est que peu importe les valeurs de kp,ki et kd

j'ai toujours des oscillation qui dépasse et une monter assez lente même si dans la

configuration de la bibliothèque j'ai réduit le temps d'échantillonnage mais que dalle il met du temps a ce stabilise et il parvient à faire des pic de +80°C. que faire svp et merci

[Répondre](#)



Chris dit :

16 mai 2023 à 22h35

Bonsoir Ness,

Une fenetre plus petite ne fait qu'augmenter la fréquence des oscillations. 5000 ou 5 secondes n'est pas trop pour du chauffage. La premiere chose a verifier est que la co.signe et la mesure de temperature son bien tous les deux en degrés. setpoint et Input. Il vous faut donc corriger la mesure en fo.ctiobde votre courbe de thermocouple. Ensuite, pour definir kp, il faut avoir quel effet a un allumage de la resistance pendant une milliseconde sur la temperature. Divisez ce deltat par le temps desire pour atteindre la consigne. Testez le resultat. Il doit y avoir des oscillations. Diminuer kp de moitie et definir $ki=kp/5$. Ca devrait etre plus lent mais converger. Ajouter un petit $kd=ki/10$ et voir l'effet. Apres. Augmenter/diminuer les valeurs de 5% et voir l'effet.

[Répondre](#)



Yalinder dit :

2 avril 2023 à 0h11

Bonsoir, votre tuto est vraiment super bien fait ! Cependant, mon but étant de contrôler la température d'une boîte 30*30 avec une température de consigne de 60 degrés soumise à des variations de T, je me demandais s'il était possible de remplacer le relai SSR par un générateur de courant /tension ? Si oui, à quoi ressemblerait le circuit et fonctionnera-t-il avec le code fourni ?

[Répondre](#)**loukos** dit :

17 janvier 2023 à 21h27

Bonjour,

Je galère à comprendre le fonctionnement général du PID.

A ce que je comprends votre sortie et tout ou rien. Mais je vois dans votre code :

```
myPID.SetOutputLimits(0, WindowSize);
```

Je comprends là que la sortie sera comprise entre 0 et 5000 ?

Est ce que dans ce cas la fonction retourne une durée comprise entre 0 et 5000, et vous mettez à 1 ou à 0 la sortie en fonction du temps retourné par la fonction ?

Et vous "computez" le PID à chaque itération ?

[Répondre](#)**stephane** dit :

5 mars 2023 à 0h06

Bonsoir Loukos, Peut-etre avez-vous deja répondu a ces questions ? Je ne suis pas un spécialiste mais ce que je peux dire c'est que pour réguler la puissance de sortie, beaucoup de contrôleurs PID définissent un Temps de cycle (ici 5000 ms mais qu'on peut modifier en fonction de la masse thermique) puis, en fonction du calcul PID, activer la sortie.

Si le calcul donne 58% de puissance, la sortie sera a 1 pendant $0,58 \times 5000$ ms

[Répondre](#)

**Sarah** dit :

9 décembre 2022 à 13h12

Bonjour, j'ai utilisée un code plus ou moins similaire avec la même bibliothèque mais dans les messages d'erreurs, on m'indique que la bibliothèque "pid front end" est introuvable. J'ai essayé de la télécharger mais rien n'y fait ! Est-on obligé d'avoir cett bibliothèque ? Avez vous une solution, s'il vous plaît ?

[Répondre](#)**Tutoduino** dit :

19 décembre 2022 à 19h43

Pouvez-vous copier-coller le message d'erreur exact ?

[Répondre](#)**Yann** dit :

21 novembre 2022 à 10h32

Bonjour Tutuino,

Super programme et merci pour ce tuto.

J'aurais une question, je suis moins encore qu'un débutant:

Comment faire pour remplacer le capteur décrit par un simple thermocouple type T ?

Merci par avance

[Répondre](#)**Marc** dit :



14 octobre 2022 à 14h45

Bonjour,

Merci beaucoup pour ce tutoriel.

Petite question concernant le relai, si je souhaite alimenter un système type petit heating pad en silicone (comme ceux sur des imprimantes 3Ds par exemple) qui accepte en input du 5/12V, est-ce que je peux du coup me rabattre sur les petits modules relai 5V pour arduino (raccordé à un AC/DC 12V 3A Power Supply Adapter) plutôt que les SSR présentés dans ce tutoriel (qui sont faits, si j'ai bien compris, pour recevoir et renvoyer directement du 220V via des grosses gaines).

Mon plan c'est de pouvoir contrôler&mesurer la température dans 2 boites de pétri via un système de chauffage par 2 heating pads, chacun situé en dessous d'une boite de petri (chaque boite de pétri contenant un DS18B20). Je serai ravi de pouvoir discuter de mon plan avec vous si vous en avez le temps et l'envie.

[Répondre](#)



Tutoduino dit :

15 octobre 2022 à 9h18

Bonjour,

Ce montage fonctionne avec tout type de relai. Attention toutefois à la puissance nécessaire au fonctionnement du relais. La puissance fournie par l'Arduino est limitée sur ses broches de sortie. Mais si vous utilisez un module relai 5V prévu pour l'Arduino cela sera parfait. Vous pouvez échanger avec moi via mon email indiqué sur la page contact si vous le souhaitez.

[Répondre](#)



Tutoduino dit :

15 octobre 2022 à 9h20

Voir cet article pour plus de détails sur le contrôle d'un relai par un Arduino <https://tutoduino.fr/tutoriels/controler-ventilateur-brushless-arduino/>

[Répondre](#)



Amy dit :

22 août 2022 à 21h43

Je ne peut pas disposer de relais SSR et donc j'ai un relais avec optocoupleur comment pourrais je donc faire pour obtenir une régulation et aussi j'utilise comme capteur de température dht 22

[Répondre](#)



Tutoduino dit :

23 août 2022 à 14h49

Le principe reste le même avec tout type de relais.

Concernant le capteur vous pouvez regarder ici : <https://tutoduino.fr/tutoriels/capteur-de-temperature-humidite-sans-fil/>

[Répondre](#)



Jean-Yves .Goret dit :

10 août 2022 à 13h03

Bonjour a tous

mon relai SSR ne se met jamais en fermeture totale meme quand la température est en

dessous de la consigne, ne faut til pas mettre une résistance de tirage entre la borne

d'entrée 3 +et 4- du relais ssr?

merci de vos propositions!

[Répondre](#)



robert dit :

11 août 2022 à 19h02

Pourriez-vous donner des détails sur votre montage ? Et comment voyez-vous que le relai n'est pas totalement fermé ?

[Répondre](#)



Goret dit :

5 juillet 2022 à 21h04

Bonjour

merci beaucoup pour ce tuto bien expliqué, ce serait bien d'avoir le câblage, pour ceux qui apprennent comme moi !

est ce qu'avec un relai SSR DC Ac on pourai piloter une lampe chauffante a infra rouge et faire de la régulation pid

pour obtenir une température constante ?

[Répondre](#)



Tutoduino dit :

6 juillet 2022 à 22h01

Oui ce type de montage avec un relai SSR DC AC pourrait très bien piloter une lampe chauffante à infra rouge et faire de la régulation PID pour avoir une température

constante.

Au niveau de votre schéma il est simple :

- Relier la broche 4 de l'Arduino à la broche + de l'input du relai SSR
- Relier la broche GND de l'Arduino à la broche - de l'input du relai SSR
- Relier la phase de votre alimentation à la broche - de l'output du relai SSR
- Relier le + de l'output du relai SSR à la phase de votre lampe
- Relier le neutre de votre alimentation au neutre de votre lampe

Attention à ne pas vous mettre en danger en manipulant du courant AC 220V...

[Répondre](#)



Jean-Yves Goret dit :

26 juillet 2022 à 23h34

Merci beaucoup, je vais essayer cela en faisant attention !
c'est un très bon Tutoduino

[Répondre](#)



Jean-Yves.Goret dit :

27 juillet 2022 à 16h28

J'ai fait le câblage comme vous le préconisé, cela fonctionne très bien ,il me reste
a peaufiné le PID tuning,pour ajuster au mieux
merci beaucoup, cela m'a beaucoup aidé Bonne journée.

[Répondre](#)



Tutoduino dit :

28 juillet 2022 à 17h59

Parfait, merci pour votre retour.

[Répondre](#)**Jean-Yves goret** dit :

2 août 2022 à 17h42

Bonjour

oui cela fonctionne mais c'est pas très facile pour trouver les bon $K_p, K_i, K_d!$
d'autant plus que je veux avoir plusieurs palier 30° , après 35° après 40° ,
après $45^\circ, 50^\circ$, et 55°

il faudra que je change bien sur la consigne mais est ce qu'il ne faudra pas
adapter les paramètres pid pour chaque niveau

peut on rajouter dans le programme la valeur qui entre dans le relai SSR a
partir de la pin D4?

merci de votre aide.

Laisser un commentaire

Votre adresse e-mail ne sera pas publiée. Les champs obligatoires sont indiqués avec *

Commentaire *

Nom *

E-mail *

Site web

Laisser un commentaire



- Accueil
- Les bases de l'électronique
- Le coin des débutants
- Tutoriels
- Blog
- Contact
- 

Derniers articles du blog

- Lire les données d'un capteur de temperature Xiaomi avec un Arduino Nano ESP32 via BLE 6 janvier 2024
- Découvrir Bluetooth® Low Energy (BLE) avec un Arduino Nano ESP32 30 décembre 2023
- Découvrir FreeRTOS sur un ESP32 avec PlatformIO 1 novembre 2023
- Affichage des données d'un capteur de température SI7021 via le serveur web d'un ESP32 13 juillet 2023
- Utiliser Redis pour stocker les données d'un objet connecté et Grafana pour les visualiser 2 juillet 2023
- Hébergez votre projet Arduino dans GitHub 24 juin 2023
- Introduction au PIO (Programmable Input Output) du RP2040 22 décembre 2022
- Vérifiez la réputation des adresses IP publiques avec lesquelles vos équipements communiquent 3 novembre 2022
- Comment ça marche une clé Dallas (iButton) ? 3 septembre 2022
- Test comparatif des PoE HAT Raspberry Pi 19 février 2022
- Programmez un Raspberry Pi Pico avec Visual Studio Code et PlatformIO 12 février 2022
- Test du PoE HAT UCTRONICS pour Raspberry Pi 4 5 février 2022

- Power over Ethernet (PoE) 8 janvier 2022
- Qu'est ce que... l'impédance caractéristique d'un câble Ethernet ? 26 décembre 2021
- Comment ça marche... Ethernet 20 novembre 2021
- J'ai testé... la LilyGo T-Watch 2020 V3 22 mai 2021
- A quoi ça sert... un module RTC ? 29 novembre 2020
- A quoi ça sert... un réseau de résistances R-2R ? 10 octobre 2020
- C'est quoi... un signal PWM ? 25 septembre 2020
- A quoi ça sert... un pont diviseur de tension ? 19 septembre 2020
- Comment utiliser... un écran OLED I2C 128x64 ? 13 septembre 2020
- C'est quoi... une tension de référence ? 20 août 2020
- A quoi ça sert... une résistance de rappel ? 19 août 2020

Copyright 2023 - Tutoduino