L'Interrupteur Automatique de Champ (IAC) du crocodile
Version à microcontrôleur

Version : V1.0
Date : 9 mai 2013
Auteur : croco31

Résumé :
Ce document décrit la construction d'un prototype d'un Interrupteur Automatique de Champ (IAC) qui est un circuit de déconnexion de la ligne de phase d'un circuit d'alimentation de prise ou de lampe quand celle-ci n'est pas utilisée, afin de supprimer toute pollution électrique inutile par les fils du secteur 240V/50Hz.
On décrit ici une alternative utilisant un microcontrôleur PIC12F675, ce qui permet d'économiser du câblage par rapport au modèle initial basé sur des composants discrets.

Avertissement :

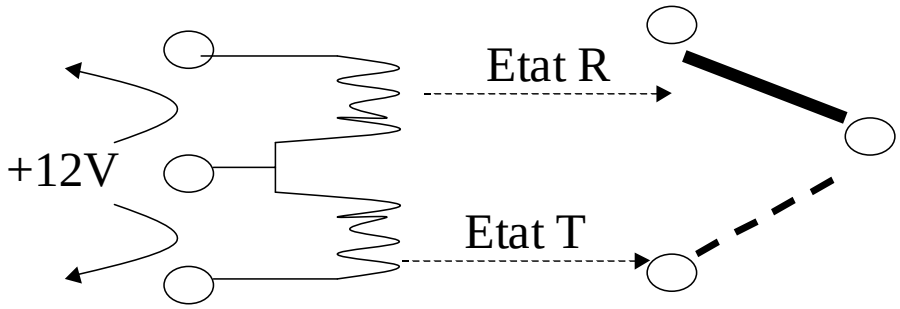
Les informations données ici sont destinées à la réalisation expérimentale d'un montage électronique. L'auteur met en garde contre le risque électrique de la manipulation d'un circuit relié directement au secteur 240V et décline toute responsabilité suite à son usage.

1. Introduction

Les principes de base de l'Interrupteur Automatique de champ (IAC) sont déjà décrits, je n'y reviens pas.

2. Choix du relais

On utilise le même relais que dans le montage à composants discrets.



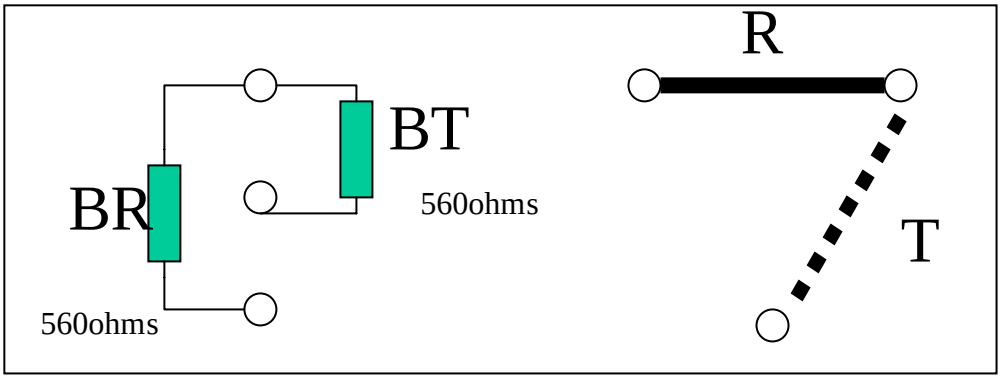
Relais bistable 1RT à 2 bobines 12V

Le modèle de contacts doit être un 1RT (RT= Repos/Travail) au moins : un point est relié à un point1 dans un état et à un point 2 dans l'autre état du relais. Si c'est un 2RT on peut relier les contacts en parallèle pour avoir plus de courant admissible (par exemple un 2RT de 8A/250V est OK pour tenir le courant 16A/250V).

J'ai retenu 1modèle de relais pour circuit imprimé que j'avais en stock (15euros HT):

- ADJ14012 de Panasonic (Ref RS :699-5768 chez Radio-Spares) : modèle bistable à 2 bobines 12V avec point commun et 1RT 16A/250V. Sous 12V chaque bobine tire 20mA environ.

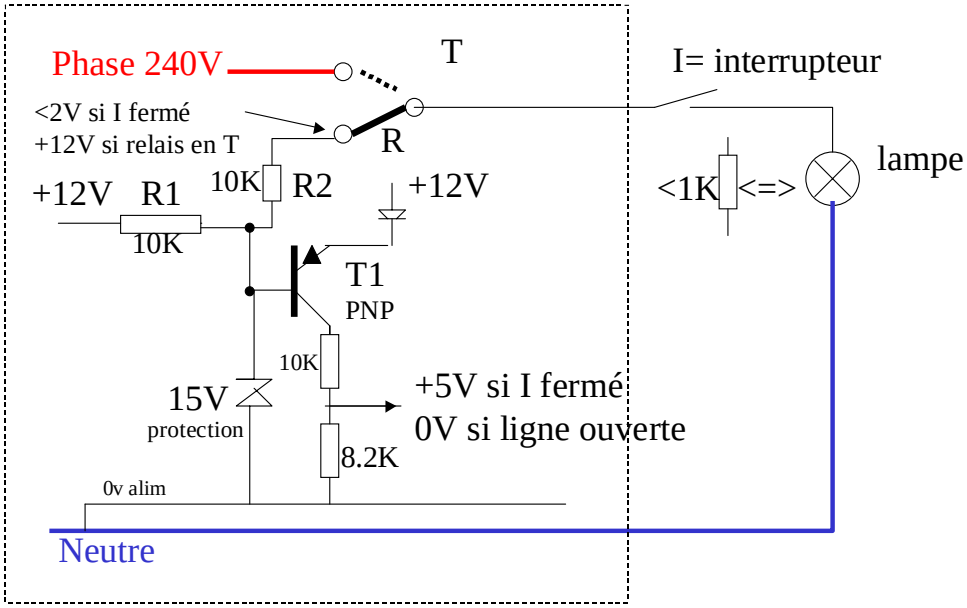
ADJ10012



Brochage vue de dessous du relais ADJ14012

3. La détection d'utilisation de la ligne de prise

La prise de ligne, c'est-à-dire l'allumage de l'interrupteur de la lampe, est détectée comme dans le montage précédent, par un transistor PNP.



Principe de détection d'utilisation de la ligne

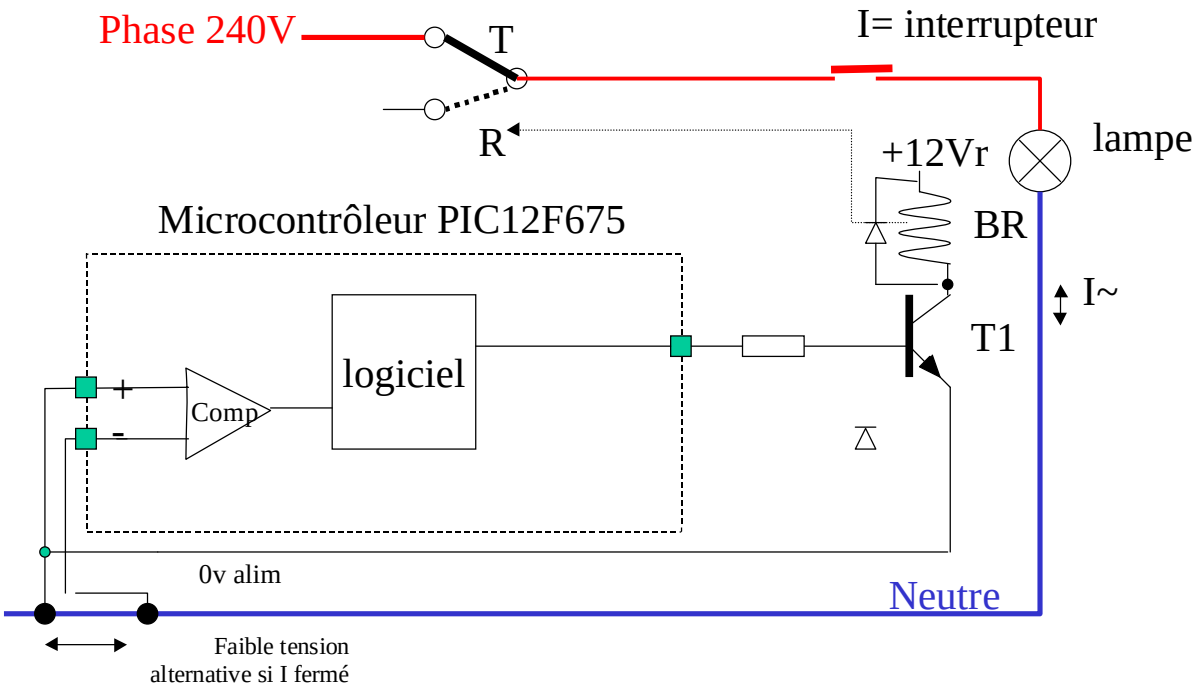
Le relais est en position R ce qui applique une tension de 12V sur la ligne de phase de la lampe. Quand l'interrupteur est fermé, le transistor devient passant, ce qui génère une tension de 5V (pas plus pour le PIC), qui sera détectée par le microcontrôleur. Au bout de 100ms environ où le logiciel détectera cette situation, ce qui protège contre des parasites, le logiciel activera une impulsion de commande du relais pour le passer en position T et appliquer ainsi la phase sur le ligne de la lampe.

4. La détection de coupure de la ligne de prise

La détection de passage d'un courant alternatif sur la ligne est effectuée directement par le microcontrôleur PIC12F675, en utilisant le comparateur interne disponible , ce qui économise un AOP externe pour faire ce travail.
La même méthode du shunt est utilisée que pour le schéma à composants discrets.

Le shunt est simplement un bout de fil de neutre aux bornes duquel le comparateur détecte une tension. La résistance du shunt dépend de la longueur de fil de neutre qui est pris dans la mesure (0.01ohms par exemple) et donne la sensibilité du montage. Cela n'a aucune dissipation (pas plus que le fil de neutre lui-même).
Attention : le 0V du montage doit être connecté sur le neutre en amont du shunt : du coté de l'arrivée du neutre. Ceci évite que le courant propre au montage soit détecté comme courant de prise, ce qui pourrait l'activer tout le temps.

Quand aucun courant alternatif ne passe, le logiciel voit un état de sortie du comparateur stable à 0 ou 1 en fonction de la tension d'offset sur les entrées et du courant continu qui peut passer.



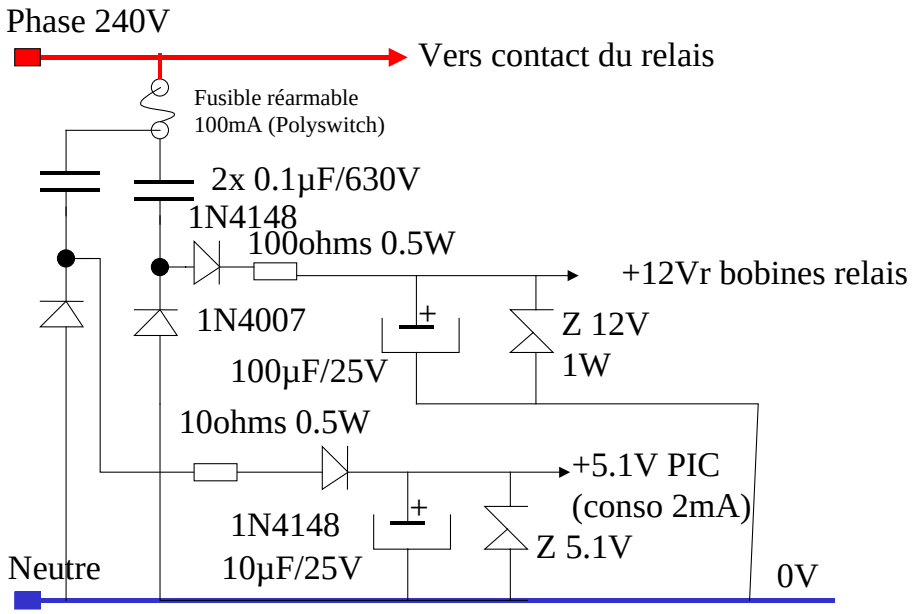
Principe de détection de coupure de la ligne

Il peut arriver que suivant la longueur du shunt et la tension d'offset résiduelle du comparateur interne du PIC, il faille inverser les broches CIN+ et CIN- du comparateur. Cette tension d'offset peut faire croire au logiciel qu'aucun courant alternatif ne traverse le shunt , car la sortie du comparateur reste bloquée à 0 ou 1.

Au bout de 15 à 20 secondes environ où la sortie du comparateur est stable (ne bouge pas à 0 ou 1), le logiciel décidera que la ligne n'est pas utilisée et enverra une impulsion de 20à 30ms environ sur la bobine BR du relais.

5. Alimentation du circuit

Le circuit est alimenté sans transformateur en utilisant un condensateur de chute de 0.1µF/630V pour chacune des tensions +12V et +5V nécessaires au PIC et au relais.
On aurait pu aussi utiliser un régulateur 78L05 pour obtenir la tension 5V du PIC à partir du 12V, mais le régulateur demande 2 à 3mA pour lui-même , soit plus que le PIC lui-même.
Une tension négative de -0.4V avait été aussi prévue par rapport au neutre, pour que le comparateur interne du PIC ait une tension de mode commun au dessus de 0V, mais la pratique a montré que cela n'est pas nécessaire si on connecte bien un coté du shunt sur le 0V directement.



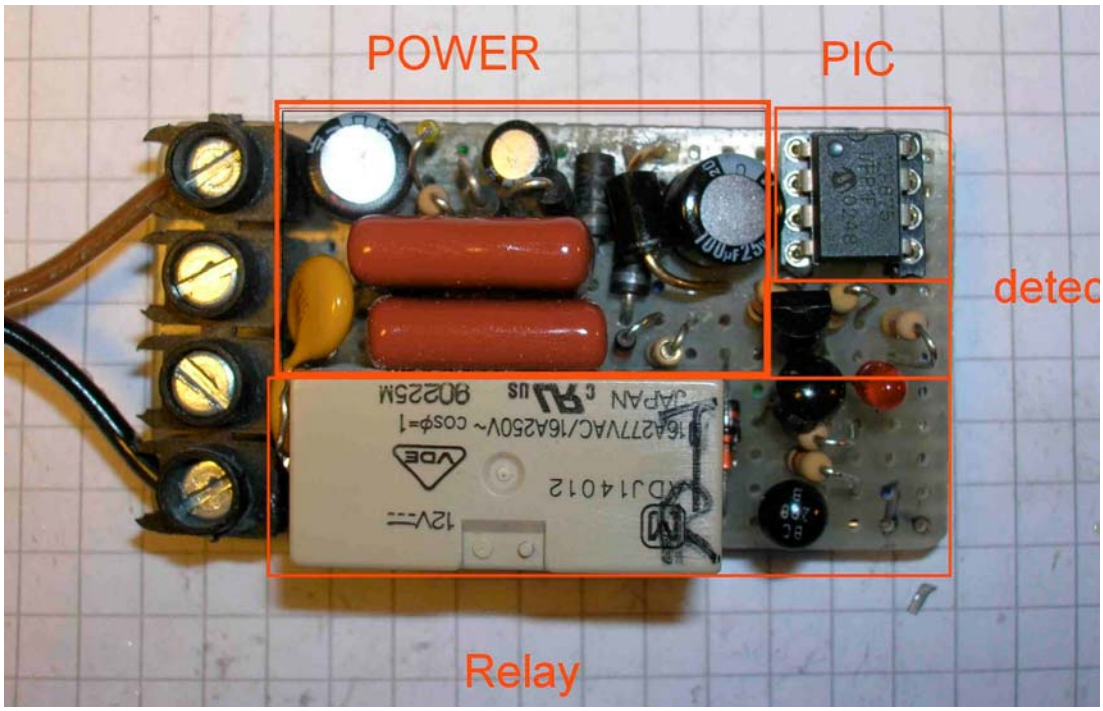
Alimentation directe (12V relais indépendante)

6. Réalisation

Le circuit final a été câblé sur une plaquette à pastilles en epoxy, en y allant progressivement, car les composants sont montés debout et serrés. La plaquette a été coupée à la fin à la bonne longueur.
Le bloc alim avec le bornier de connexion et le relais a d ‘abord été monté et testé en vérifiant sur secteur (DANGER) que les tensions +12Vr et Vcc= 10.4V étaient bien présentes.

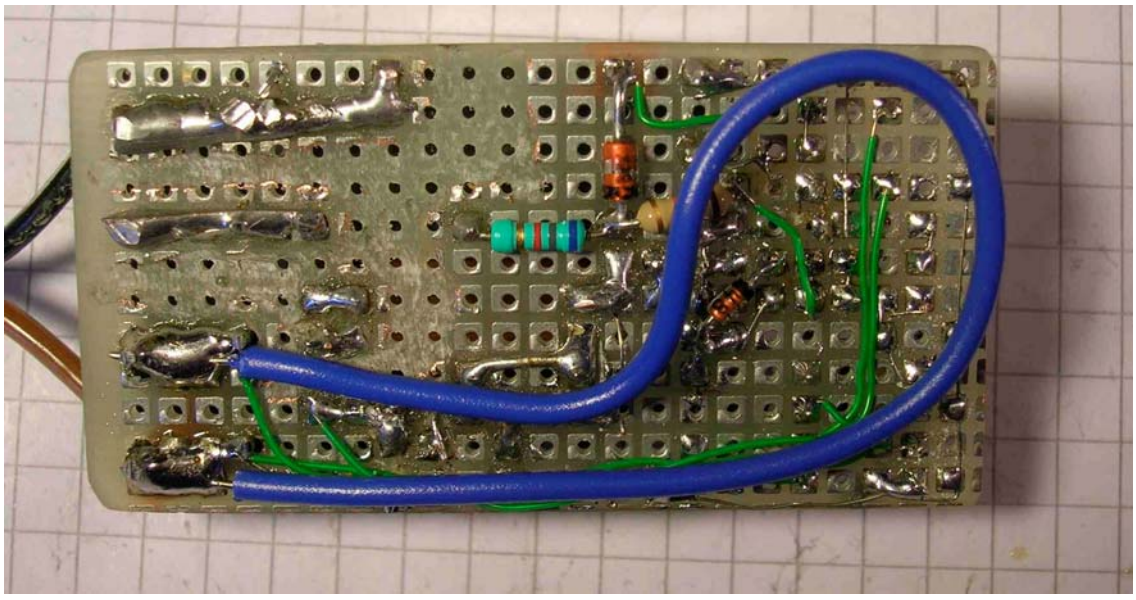
Puis le bloc de détection de prise de ligne a été monté et testé en alimentant par une alim labo aux bornes de la zener de 12V (attention : limiter le courant de l’alim à 10mA ou la tension à 11.5V sous peine de tuer la zener), et en simulant la prise de ligne par une résistance de 1Kokms entre PhaseOUT et neutre : on voit bien la tension de détection passer de 0V à 5V environ (pas plus). Les transistors de commande du relais ont ensuite été montés, en testant l’activation de la bobine par une résistance au 12V sur la 10K de commande de chaque transistor.

Enfin le PIC a été monté sur support, ainsi que la LED faible consommation (max 2mA) , celle-ci est bien pratique pour vérifier que le logiciel démarre , car il envoie plusieurs flashs sur le LED au démarrage. De plsu à chaque activation de relais, des flashes sont aussi émis sur le LED.



Enfin le shunt du neutre a été monté coté pastilles : les broches du comparateur interne du PIC vont se prendre directement par deux fils sur le bornier d'entrée du Neutre.

De plus les pastilles inutilisées autour du bornier on été supprimées pour avoir une bonne isolation entre phase et neutre.



Sa taille est 60x35x25mm environ, soit 2cm de moins que le module à composants discrets.

Les essais ont ensuite été faits sur table en le connectant sur secteur :

- il détecte bien une ampoule à filament de 28W : si on coupe l'ampoule, l'IAC déconnecte la phase au bout de 15 secondes environ.
- Il détecte bien une ampoule à LED de 4W

7. Logiciel

Le logiciel du PIC12F675 est développé en C (compilateur PICC Hitech-C de HTSOFT, dont une version Lite gratuite peut être chargée depuis le site www.htsoft.com).

Le PIC 12F675 est configuré pour utiliser l'oscillateur interne 4MHz, et utilsier toutes ses broches en broches IOs.

Il se programme à partir du fichier .HEX qui contient aussi les bits de configuration du PIC.

En voici une copie (c'est du texte) :

```
:10000000830100308A00042821308400263011202A
:1000100020308400213011208301482B0406800108
:0A002000840A0406031D0E280034B4
:10064E008312A10183160512831205161E303C2358
:10065E00051208008312A101A10A831685128312C6
:10066E0085161E303C23851208008312A400FA3032
:10067E00A500422B6400A503250F412BA40B3E2B96
:10068E000800643083123C236400FF30831681001F
:10069E00831205118316051185120512831285121 8
:1006AE00051283160514851 41F149F141F119F1114
:1006BE008312272383168E14643083123C23051570
:1006CE0032303C23051132303C23051532303C23A9
:1006DE00051164303C238312A108031DDC2B02306C
:1006EE009900051183160511103083 12A2002730D0
:1006FE00A300201023082204031 9C42B6400013028
:10070E0083123C23A010191BA0148312A01C912B42
:10071E000515922B0511201C962B0130972B0030BE
:10072E00A600A01C9C2B01309D2B0030A606031D9D
:10073E00A52BA2080319A303A203812B2010A01C32
:10074E00A92B201423308312230228300319220 2EE
:10075E000318B82BE830A2070318A30A0330A30727
:10076E00812B27302302063003192202031 8812B16
:10077E000A30A2070318A30A812B05113230831207
:10078E003C23051532303C23051132303C23051530
:10079E0032303C23051132303C23051532303C23D8
:1007AE000511312383160515722B83160515831239
:1007BE00A201A30183161F11003083122302643 09D
:1007CE00031922020318F72B640001303C23051D88
:1007DE00F42BA20A0319A30AE32BA201A301E32B14
:1007EE000511272364303C23051D722B2723722B02
:02400E00CC3FA5
:00000001FF
```

Le logiciel est très simple :

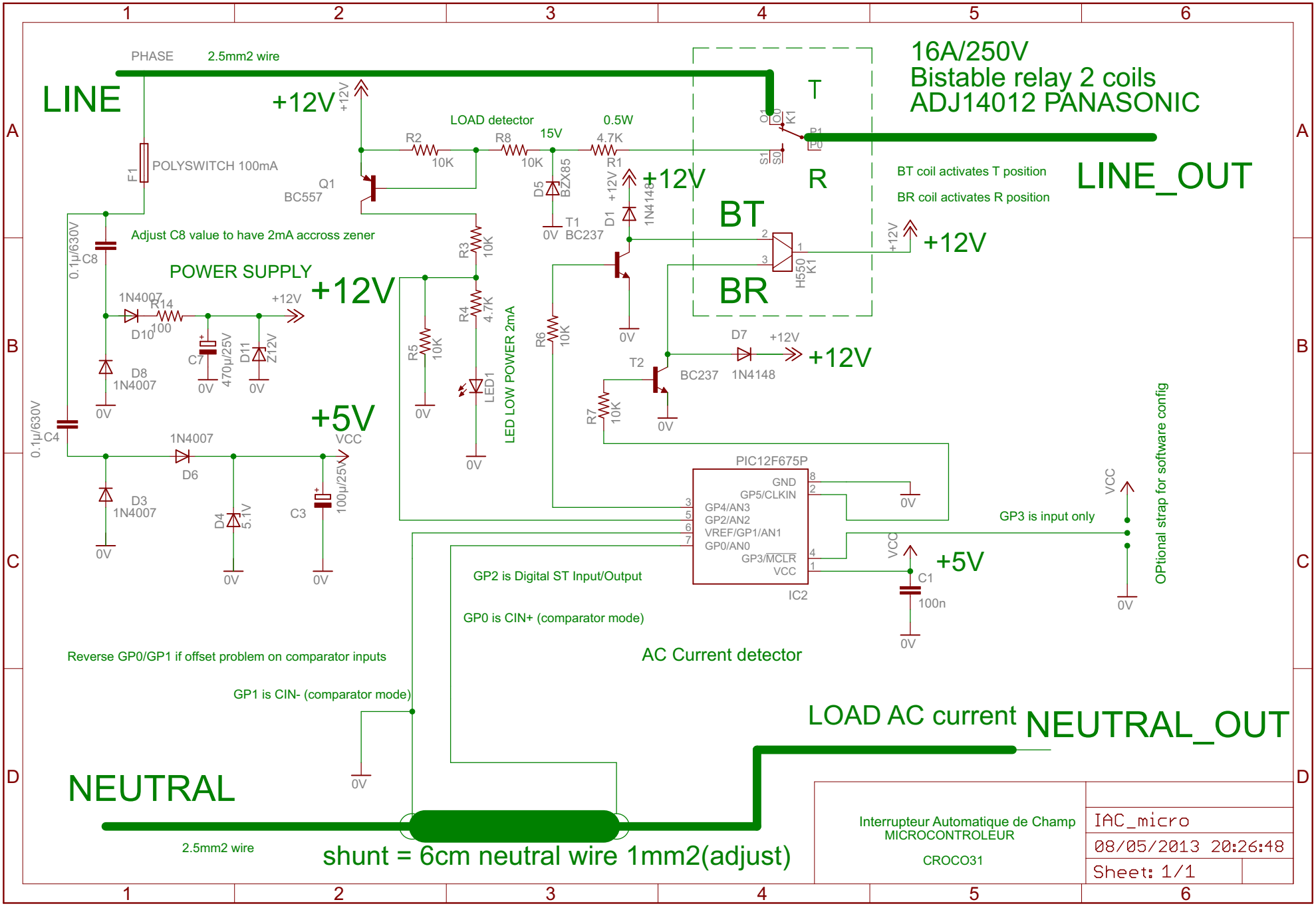
- une initialisation (main) configure les registres internes du PIC12F675 et fait plusieurs pulses sur la LED qui est branchée sur le port GP2.
- Le logiciel active par défaut la bobine T du relais au démarrage : la phase est activée par défaut après une coupure du secteur et on passe dans l'état PHASE_ON
- Il boucle ensuite entre 2 états :
  - L'état PHASE_ON : dans cet état le changement de la sortie du comparateur interne indique qu'un courant alternatif t traverse le shunt : on passe à l'autre état de repos quand la sortie est stable pendant au moins 20 secondes environ. La LED clignote au moment de changer d'état et appliquer une impulsion de 30ms environ sur la bobine BR. Dans l'état PHASE_ON, l'état du bit de sortie du comparateur est affiché sur le diode LED sur GP2 qui est en sortie.
  - L'état PHASE_OFF : dans cet état le port GP2 est mis en entrée et son état est surveillé. S'il passe à 1 pendant plus de 100ms environ, on repasse dans l'état PHASE_ON en appliquant une impulsion sur la bobine BT du relais.
- le Watchdog interne est actif et provoquera un reset du PIC en cas de problème.

Le source C :

```
/* ------------------------------------------------------------------*/
/* Prog de controle de l'Interrupteur Automatique de Champ du crocodile */
/* Pour PIC12F675     Compilateur HiTech C  PICC V8.02                */
/* Auteur: croco31                                                    */
/* ------------------------------------------------------------------*/
/* Modifs:
   04/04/2013  JLC creation
   28/04/2013  JLC modification detection de charge par transistor PNP
 */

/* uses pic12f6x.h */
#include <pic.h>


/*
Configuration bits of PIC12F675
 - internal oscillator 4MHz
 - GP4 and GP5 are used as IO pins
 - WDG active
 - Power Up Timer active
 - MCLR used as IO
*/
__CONFIG(INTIO & UNPROTECT & WDTEN & PWRTEN & MCLRDIS);


static char Relay_state=0; /* current State of automaton and relay 0= PHASE_ON 1= PHASE_OFF */
/*
 Pin usage of PIC12F675:
 -----------------------
  VSS  (8): GND connected to -0.4V voltage to cope with the common mode range of the comparator (0V to VDD-
1.5V)
  GP0  (7): IO input+ of the comparator for detecting LOAD current on neutral line
  GP1  (6): IO input- of the comparator for detecting LOAD current on neutral line
  GP2  (5): IO input (Schmitt trigger) for detecting the lamp switch activation + LED output when phase
active
  GP3  (4): INPUT ONLY: reserved for optional  mode setting
  GP4  (3): IO output command of the BT coil of the relay  (put the PHASE onto the line)
  GP5  (2): IO output command of the BR coil of the relay  (remove PHASE from the line)
  VDD  (1): power supply +5V

*/

/* definition of signals onto ports */
/* ------------------------------- */
#define pLamp_detect GPIO2
#define pLED         GPIO2
#define tLamp_detect TRIS2
#define pBTcoil (GPIO4)
#define tBTcoil (TRIS4)
#define pBRcoil (GPIO5)
#define tBRcoil (TRIS5)
#define tComp0  TRIS0
#define tComp1  TRIS1
/* GP3 input is used for mode setting */
#define pMode   GPIO3
#define tMode   TRIS3

/* control of LED */
/* LED is ON when GP2 is an output and at +5V */
#define LED_IS_ON 1
#define LED_IS_OFF 0
/* detect lamp switch on GP2=1 */
#define DETECT_LAMP_ON 1

fastcall void
Delayms (unsigned char nbmilli)
{
unsigned char count;
       do {
         count=250;
         while(count--) {asm("clrwdt");}
       } while (--nbmilli);
}


/* activates PHASE onto the output wire */
/* ---------------------------------- */
void
Set_phase_ON(void)
{
```

```
 Relay_state=0;
 /* 30ms pulse onto the BT coil of relay */
 tBTcoil=0; /* confirm port GP4 is output */
 pBTcoil=1;
 Delayms(30); /* pulse 30ms */
 pBTcoil=0;
}

/* deactivates PHASE onto the output wire */
/* ------------------------------------ */
void
Set_phase_OFF(void)
{
 Relay_state=1;
 /* 30ms pulse onto the BR coil of relay */
 tBRcoil =0; /* confirm output port */
 pBRcoil=1;
 Delayms(30); /* pulse 30ms */
 pBRcoil=0;
}



/**********************/
/* Main program       */
/**********************/

main (void)
{
unsigned int count;
static bit save_cout,new_cout;

 Delayms(100); /* 100ms delay at startup */
 asm("clrwdt"); /* clearing internal watchdog */

 /* configure internal watchdog period */
 /* ---------------------------------- */
 OPTION= 0xFF; /* WDT prescaler = 128 */

 /* PORTs initialisation */
 /* -------------------- */
 pLED=         LED_IS_OFF; /* LED is OFF */
 tLamp_detect= 0; /* GP2 set as output to command LED  */
 tBRcoil=      0; /* output */
 tBTcoil=      0; /* output */
 pBRcoil=      0; /* coil BR is OFF */
 pBTcoil=      0; /* coil BT is OFF */
 tComp0=       1; /* GP0 is INPUT */
 tComp1=       1; /* GP1 is INPUT */

 /* configure internal comparator on GP0(CIN+) and GP1(CIN-) */
 /* ---------------------------------------------------------*/
 ANS0 = 1;    /* GP0 is analog input */
 ANS1 = 1;    /* GP1 is analog input */
 ANS2 = 0;    /* GP2 is digital port */
 ANS3 = 0;    /* GP4 = AN3 is digital port */

 /*  POWER ON     set Relay to BT position : PHASE is ON by default at reboot   */
 /*  because tehre is no way for the PIC to know the relay state  R or T        */
 /* ------------------------------------------------------------------------- */
 Set_phase_ON();
 POR=1; /* mark power up is done */

 /* Blink LED 2 times pulses 50ms at startup */
 /* -------------------------------- */
 Delayms(100);
 pLED=LED_IS_ON; /* LED ON */
 Delayms(50);
 pLED=LED_IS_OFF; /* LED OFF */
 Delayms(50);
 pLED=LED_IS_ON; /* LED ON */
 Delayms(50);
 pLED=LED_IS_OFF; /* LED OFF */
 Delayms(100);

 /* automaton loop */
 /* -------------- */
 while(1) {

  if( Relay_state==0) {
    /* Phase ON state */
    /* -------------- */
```

```
    CMCON= 0x02; /* bit COUT is 1 when GP0>GP1 and 0 otherwise*/
    pLED= LED_IS_OFF; /* LED is OFF */
    tLamp_detect = 0; /* GP2 is output for LED control */
    /* Test absence of AC current: COUT must be stable during at least 10 seconds */
    count=10000;
    save_cout= 0;
    while(count!=0) {
     asm("clrwdt"); /* clearing internal watchdog */
     Delayms(1); /* sample COUT every 1ms for 50Hz or 60Hz */
     new_cout= COUT;
     pLED= new_cout; /* copy COUT state onto LED to monitor current */
     /* check COUT change */
     if( new_cout==save_cout) {
      /* no change */
      count--; /* count number of milliseconds with no change */
     } else {
      /* toggle */
      save_cout= new_cout;
      if( count<9000) {
       count+=1000;
      } else {
       if( count <9990) count+=10;
      }

     }
    }
    /* ---- Absence of AC current is detected: activates BR coil to remove PHASE */
    pLED= LED_IS_OFF; /* LED is OFF */
    Delayms(50); /* 3 pulses on LED */
    pLED= LED_IS_ON;  /* LED is ON */
    Delayms(50);
    pLED= LED_IS_OFF; /* LED is OFF */
    Delayms(50); /* 3 pulses on LED */
    pLED= LED_IS_ON;  /* LED is ON */
    Delayms(50);
    pLED= LED_IS_OFF; /* LED is OFF */
    Delayms(50); /* 3 pulses on LED */
    pLED= LED_IS_ON;  /* LED is ON */
    Delayms(50);
    pLED= LED_IS_OFF; /* LED OFF */

    Set_phase_OFF();
    tLamp_detect = 1; /* GP2 becomes digital input */

  } else {
    /* Phase OFF state */
    /* --------------- */
    tLamp_detect = 1; /* GP2 is digital input */
    count = 0;
    /* Test GP2 : 0 if lamp is switched on  */
    /* GP2 is digital input with Schmitt trigger */
    ANS2= 0; /* set digital */
    while(count<100) {
     asm("clrwdt"); /* clearing internal watchdog */
     Delayms(1);
     if( pLamp_detect == DETECT_LAMP_ON) { /* lamp line is switched */
      count++; /* count ms: LAMP detected if GP2=0 stable during 100ms */
     } else {
      count= 0;
     }
    }
    /* ---- Lamp switch-on is detected : activates BT coil to apply PHASE */
    pLED= LED_IS_OFF;
    Set_phase_ON();
    /* here the relay must be set in T position: GP2 must return  to level 0 */
    Delayms(100);
    if( pLamp_detect == DETECT_LAMP_ON) {
     /* try again */
     Set_phase_ON();
    }

  }

 }

}
```


PHASE
2.5mm2 wire
LINE
16A/250V
Bistable relay 2 coils
ADJ14012 PANASONIC
T
R
LINE_OUT
BT coil activates T position
BR coil activates R position
BT
BR
+12V
LOAD detector
0.5W
15V
4.7K
10K
R2
R8
R1
Q1
BC557
D5
BZX85
T1
BC237
D1
1N4148
F1
POLYSWITCH 100mA
Adjust C8 value to have 2mA accross zener
POWER SUPPLY
C8
0.1µ/630V
1N4007
R14
100
D10
C7
470µ/25V
D11
Z12V
D8
R3
R4
4.7K
R5
R6
LED1
LED LOW POWER 2mA
D7
T2
BC237
1N4148
R7
+5V
VCC
C4
D6
D3
D4
5.1V
C3
100µ/25V
PIC12F675P
GND
GP5/CLKIN
GP4/AN3
GP2/AN2
VREF/GP1/AN1
GP0/AN0
GP3/MCLR
VCC
IC2
GP3 is input only
OPtional strap for software config
GP2 is Digital ST Input/Output
GP0 is CIN+ (comparator mode)
Reverse GP0/GP1 if offset problem on comparator inputs
GP1 is CIN- (comparator mode)
AC Current detector
C1
100n
LOAD AC current
NEUTRAL_OUT
NEUTRAL
2.5mm2 wire
shunt = 6cm neutral wire 1mm2(adjust)
Interrupteur Automatique de Champ
MICROCONTROLEUR
CROCO31
IAC_micro
08/05/2013 20:26:48
Sheet: 1/1