

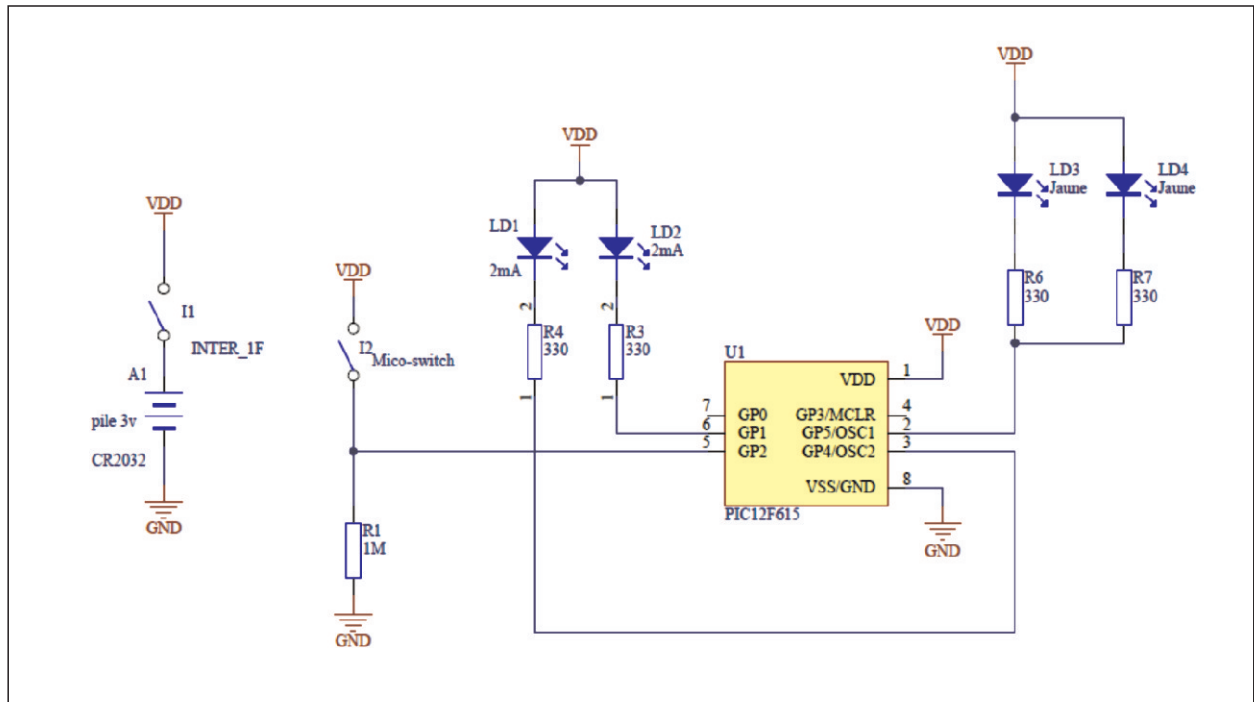


PARCOURS D'EXCELLENCE

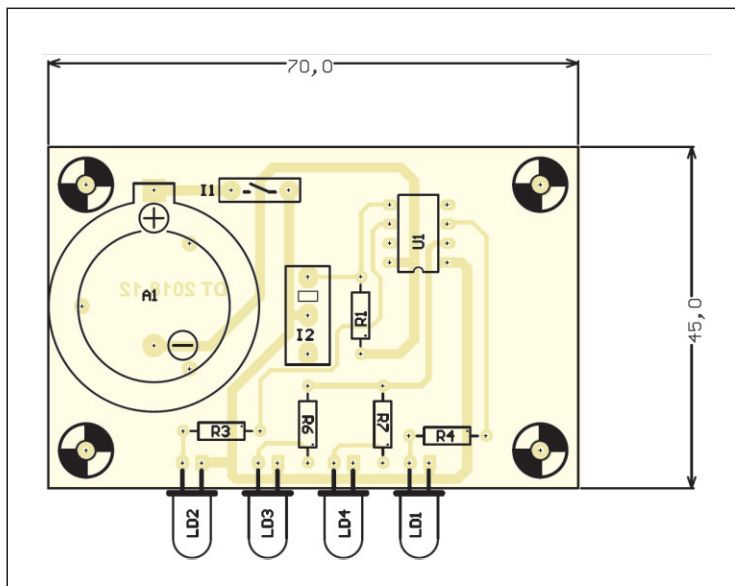


Support pour smartphone

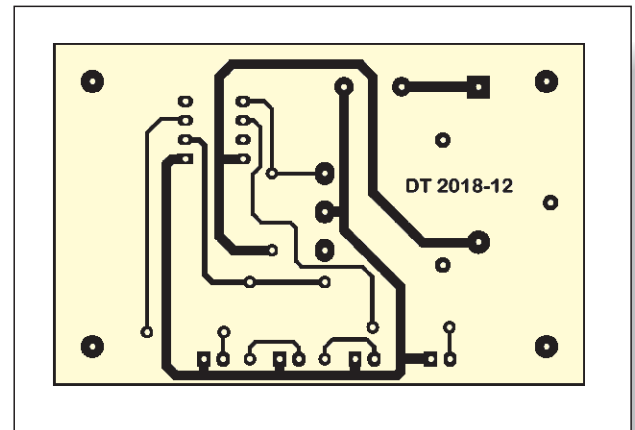
Schéma électronique



Implantation des composants



Dessin du PCB



Nomenclature

Désignation	Repère	Qté
Support de pile CR2032	A1	1
Pile 3V CR2031	A1	1
Interrupteur M/A	I1	1
Micro-switch	I2	1
Led	LD1-LD2-LD3-LD4	4
Résistance 1MΩ	R1	1
Résistance 330Ω	R3-R4-R6-R7	4
Support CI DIP8	U1	1
Microcontrôleur PIC12F615	U1	1



```

/* =====

Projet parcourt d'excellence collège

Lampe sur support smartphone

Mars 2018

===== */

//pragma chip PIC12F615
#include < PC-12F615.h"
#include <xc.h>

#pragma config CP    = 1 // Flash Program Memory Code Protection bit (Code protection off)
#pragma config BOREN  = 3 // Brown-out Reset Enable bit (BOR disabled)
#pragma config PWRTE  = 1 // Power-up Timer Enable bit (PWRT disabled)
#pragma config WDTE   = 0 // Watchdog Timer Enable bit (WDT disabled)
#pragma config IOSCFs = 1 // Internal Oscillator Frequency Select bit: Fosc = 8Mhz
#pragma config FOSC   = 4 // RC oscillator: CLKOUT function on GP4/OSC2/CLKOUT pin, RC on GP5/OSC1/CLKI
#pragma config MCLRE  = 0 // MCLRPin Function Select bit

#define _XTAL_FREQ 800000

//***** Alias (configuration physique)*****

#define led1    GPIO4 // sortie led couleur 1
#define led2    GPIO1 // sortie led couleur 2
#define led34   GPIO5 // sortie led blanche

#define Contact GPIO2 // entrée contact allumant lampe

void tempo_500us (unsigned int T_voie, unsigned short T_duree) ;
void led_clignot (int mode, unsigned short C_duree ) ;
void fontionnement (void) ;

// ----- Définition Variables -----

bit Transitors ; // état transistor : false = Tr1 / true = Tr2
bit depart ; // lance le systeme

unsigned int tempo_D1 ; // duree tempo 1 - gestion Clignotement led couleur
unsigned int tempo_D2 ; // duree tempo 2 - gestion temps led blanche allumée
unsigned int tempo_D3 ; // duree tempo 3 - gestion temps allumage led couleur
unsigned int tempo_C1 ; // compteur tempo 1
unsigned int tempo_C2 ; // compteur tempo 2
unsigned int tempo_C3 ; // compteur tempo 3
bit tempo_E1 ; // Etat tempo 1
bit tempo_E2 ; // Etat tempo 2
bit tempo_E3 ; // Etat tempo 3
bit tempo_I1 ; // inpultion de fin tempo 1
bit tempo_I2 ; // inpultion de fin tempo 2
bit tempo_I3 ; // inpultion de fin tempo 3

char sequence ; // les etapes de fonctionnement

// ----- programme principale -----
void main(void)
{
    ANSEL =0b00000000; // Dire quels sont les pins en analogique/digital pour le CAN
    TRISIO=0b000000100; // Déclaration des pins si entrée = 1 / sortie =0
    GPIO = 0x00;
    OPTION_REG=0b10000000;

    // ----- interruption ----- //Modifie PEIE
    INTCON=0xF0;

    TMR0 = 17 ;
    T0CS = 0 ;
    PSA = 0 ;
    T0IF = 0;
    T0IE = 1;
    INTF = 0;
    INTE = 1;
    INTEDG = 1 ; // front montant

//----- Initialisation -----
    depart = 0 ;
    led1 = 1 ;
    led2 = 1 ;
    led34 = 1 ;
    sequence = 0 ;

while (1) //boucle infinie
    { fontionnement () ; }

}

// ----- gestion des sequences -----
void fontionnement (void)
// 0 : arret
// 1 : eclairage
// 2 : clignotement led1
// 2 : clignotement led2
// 2 : clignotement led1 et led2 simultanees
// 2 : clignotement led1 et led2 alternees
{
    switch (sequence)
    {
        case 0 :
        {
            led34 = 1 ;
            led_clignot (0,0) ;
            if (depart == 1 )
            {
                tempo_500us (1,40000) ; // temporisation 10s
                sequence++ ;
                tempo_500us (3,1000) ;
            }
        }
        break ;

        case 1 :
        {
            led34 = 0 ;
            led_clignot (0,0) ;
            if (tempo_I1 == 1 )
            {
                tempo_500us (1,20000) ; // temporisation 10s
                sequence++ ;
                tempo_500us (3,1000) ;
            }
        }
        break ;
    }
}

```

```

case 2 :
{
    led34 = 0 ;
    led_clignot (1,1000) ;
    if (tempo_I1 == 1 )
    {
        tempo_500us (1,20000) ;           // temporisation 5s
        sequence++ ;
        tempo_500us (3,1000) ;
    }
}
break ;

case 3 :
{
    led34 = 0 ;
    led_clignot (2,1000) ;
    if (tempo_I1 == 1 )
    {
        tempo_500us (1,20000) ;           // temporisation 5s
        sequence++ ;
        tempo_500us (3,1000) ;
    }
}
break ;

case 4 :
{
    led34 = 0 ;
    led_clignot (3,1000) ;
    if (tempo_I1 == 1 )
    {
        tempo_500us (1,20000) ;           // temporisation 5s
        sequence++ ;
        tempo_500us (3,1000) ;
    }
}
break ;

case 5 :
{
    led34 = 0 ;
    led_clignot (4,1000) ;
    if (tempo_I1 == 1 )
    {
        tempo_500us (1,20000) ;           // temporisation 5s
        sequence++ ;
    }
}
break ;

case 6 :
{
    led_clignot (5,600) ;
    if (tempo_I1 == 1 )
    {
        tempo_500us (1,15000) ;           // temporisation 5s
        sequence++ ;
    }
}
break ;

case 7 :
{
    led34 = 1 ;
    led_clignot (4,1500) ;
    if (tempo_I1 == 1 )
    {
        tempo_500us (1,10000) ;           // temporisation 5s
        sequence++ ;
    }
}
break ;

case 8 :
{
    led34 = 1 ;
    led_clignot (0,0) ;
    sequence = 0 ;
    depart = 0 ;
    SLEEP () ;
}
break ;
}
}

// ----- gestion des leds couleur -----
void led_clignot (int mode, unsigned short C_duree )
// 0 = arret
// 1 = led 1
// 2 = led 2
// 3 = led1 et led2 simultanees
// 4 = led1 et led2 alternes
// 5 = led34
{
    switch (mode)
    {
        case 0 :           // arret
        {
            led1 = 1 ;
            led2 = 1 ;
        }
        break ;
        case 1 :           // uniquement led1
        {
            led2 = 1 ;
            if (tempo_I3 == 1 )
            {
                led1 = 1^led1 ;
                tempo_500us (3,C_duree) ;
            }
        }
        break ;
        case 2 :           // uniquement led2
        {
            led1 = 1 ;
            if (tempo_I3 == 1 )
            {
                led2 = 1^led2 ;
                tempo_500us (3,C_duree) ;
            }
        }
        break ;
        case 3 :           // led1 et led2 simultanees
        {
            if (tempo_I3 == 1 )

```

```

{
    led1 = 1^led1 ;
    led2 = led1 ;
    tempo_500us (3,C_duree) ;
}
}
break ;
case 4 :           // led1 et led2 alternes
{
    if (tempo_I3 == 1 )
    {
        led1 = 1^led1 ;
        led2 = 1^led1 ;
        tempo_500us (3,C_duree) ;
    }
}
break ;
case 5 :           // uniquement led34
{
    led1 = 1 ;
    led2 = 1 ;
    if (tempo_I3 == 1 )
    {
        led34 = 1^led34 ;
        tempo_500us (3,C_duree) ;
    }
}
break ;
}
}
// -----
// Gestion tempo
// T_voie = numero de la tempo si = 0 => toutes les tempos à arret
// T_duree = X * 10ms (jusqu'a 10mn 55s 350ms) si = 0 => arret

void tempo_500us (unsigned int T_voie, unsigned short T_duree)

{
    switch (T_voie)
    {
        case 0 :
        {
            tempo_C1 = 0 ;           // initialisation compteur
            tempo_C3 = 0 ;           // initialisation compteur
            tempo_E1 = 0 ;           // etat tempo = 0
            tempo_I1 = 0 ;           // Impulsion de fin = 0
            tempo_E3 = 0 ;           // etat tempo = 0
            tempo_I3 = 0 ;           // Impulsion de fin = 0
        }
        break ;

        case 1 :
        {
            if (T_duree == 0)
            {
                tempo_E1 = 0 ;           // etat tempo = 0
                tempo_I1 = 0 ;           // Impulsion de fin = 0
            }
            else
            {
                tempo_D1 = T_duree ;           // durée de la tempo
                tempo_C1 = 0 ;           // initialisation compteur
                tempo_I1 = 0 ;           // Impulsion de fin = 0
                tempo_E1 = 1 ;           // etat tempo = 1
            }
        }
        break ;

        case 3 :
        {
            if (T_duree == 0)
            {
                tempo_E3 = 0 ;           // etat tempo = 0
                tempo_I3 = 0 ;           // Impulsion de fin = 0
            }
            else
            {
                tempo_D3 = T_duree ;           // durée de la tempo
                tempo_C3 = 0 ;           // initialisation compteur
                tempo_I3 = 0 ;           // Impulsion de fin = 0
                tempo_E3 = 1 ;           // etat tempo = 1
            }
        }
        break ;
    }
}

// ----- Gestion Interruptions -----

void interrupt tc_int(void)

{
    if (TOIF == 1)
    {
        TOIF = 0 ;
        TMR0 = 17 ;

        if (tempo_E1 == 1 )
        {
            if (tempo_C1++ == tempo_D1)           // compteur tempo 1
            {
                tempo_E1 = 0 ;
                tempo_I1 = 1 ;
            }
        }

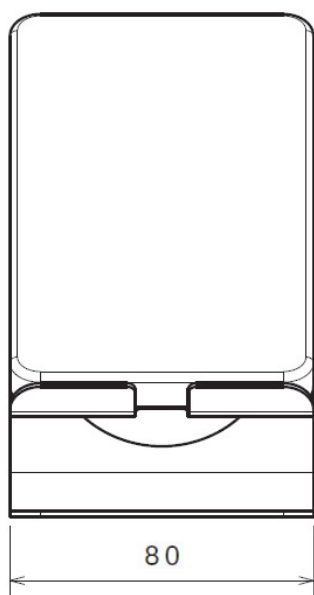
        if (tempo_E3 == 1 )
        {
            if (tempo_C3++ == tempo_D3)           // compteur tempo 3
            {
                tempo_E3 = 0 ;           //
                tempo_I3 = 1 ;           //
            }
        }
    }

    if (INTF == 1)
    {
        INTF = 0 ;
        depart = 1 ;

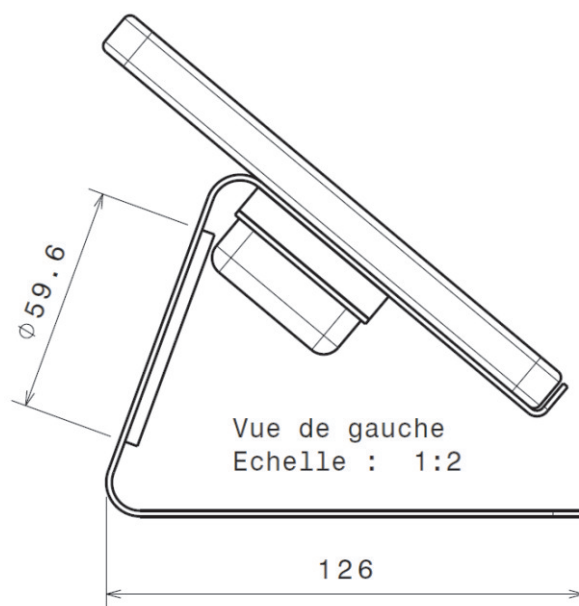
        if ( INTEDG == 0 ) { INTEDG = 1 ; } // front montant
        else { INTEDG = 0 ; } // front descendant
    }
}

// ----- fin de fichier -----

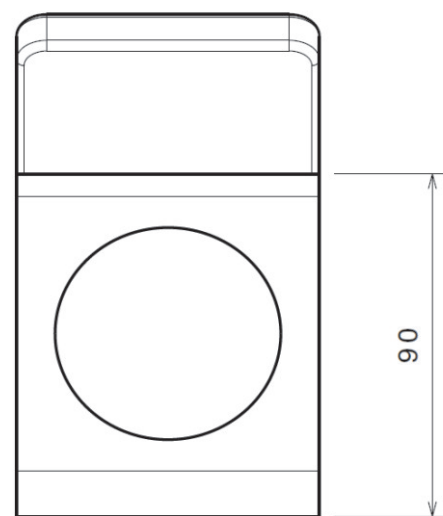
```



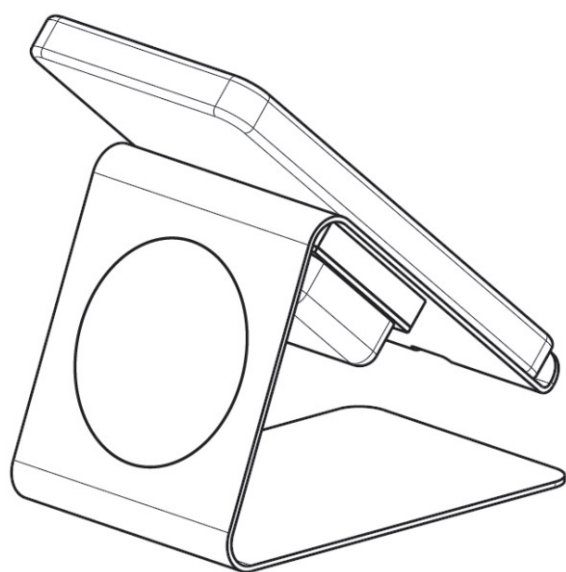
Vue de face
Echelle : 1:2



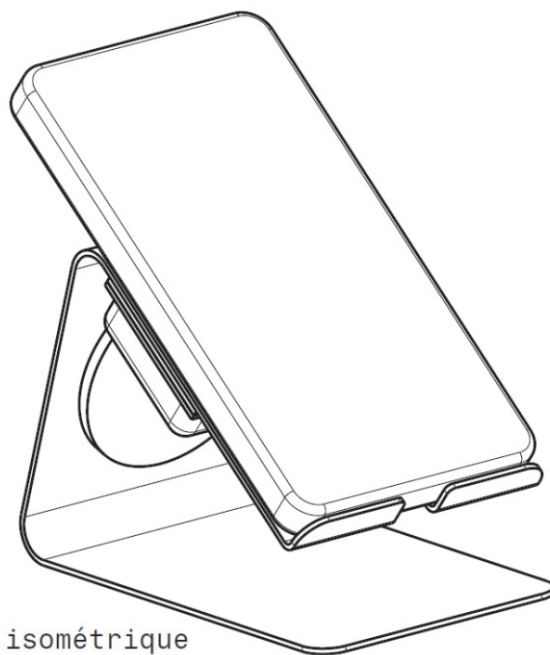
Vue de gauche
Echelle : 1:2



Vue de derrière
Echelle : 1:2



Vue isométrique
Echelle : 1:2



Vue isométrique
Echelle : 1:2

