L4 / Driver de LED

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

MOTS-CLEFS

+ Transistor
+ Miroir de courant
+ LED de puissance

COMPOSANTS

+ LED
+ Transistor
+ Drivers de LED (AL5809 et NCR320U)

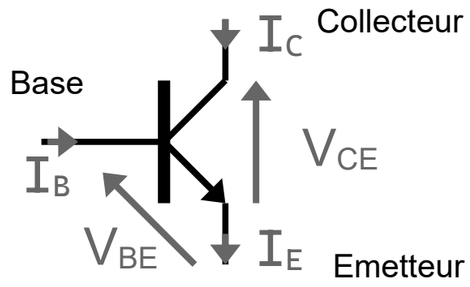
POUR COMMENCER

Sur le montage 1 :
- calculez I_{C2} en fonction de I_P
- calculez la puissance dissipée par la résistance R_P

Sur le montage 2 (composant NCR320U) :
- calculez le courant Iout et précisez l'intérêt de Rext
- calculez le courant Ien et précisez l'intérêt de Ven

Composant AL5809 :
- expliquez le fonctionnement de ce composant

Transistor Bipolaire



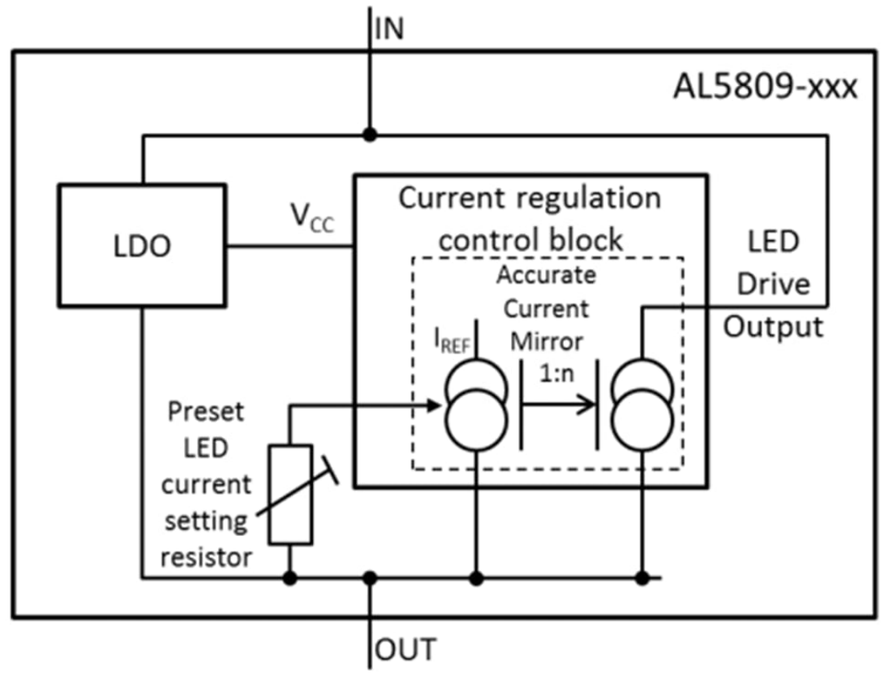
$$I_C = \beta . I_B$$

$$I_E = I_C + I_B$$

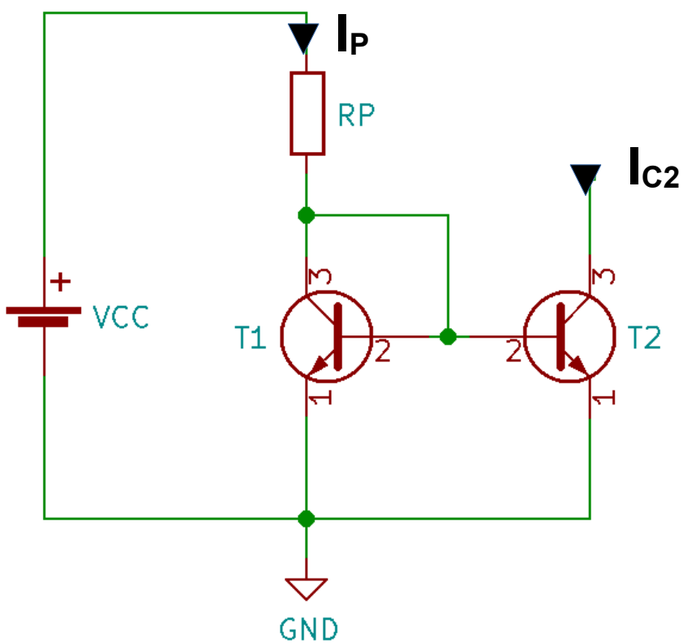
$$I_C = \beta . I_{BS} \exp(V_{BE}/U_T)$$

U_T, I_{BS} et β sont des paramètres intrinsèques du transistor

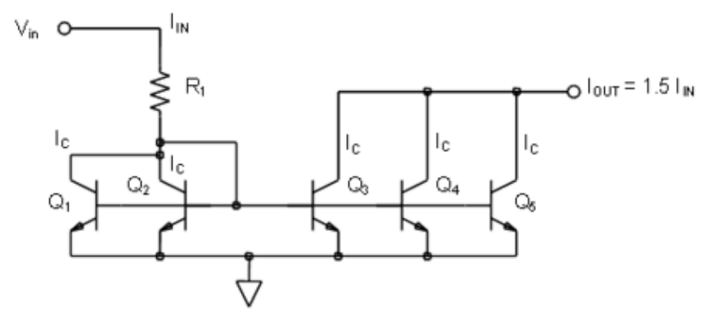
Composant AL5809



Montage 1

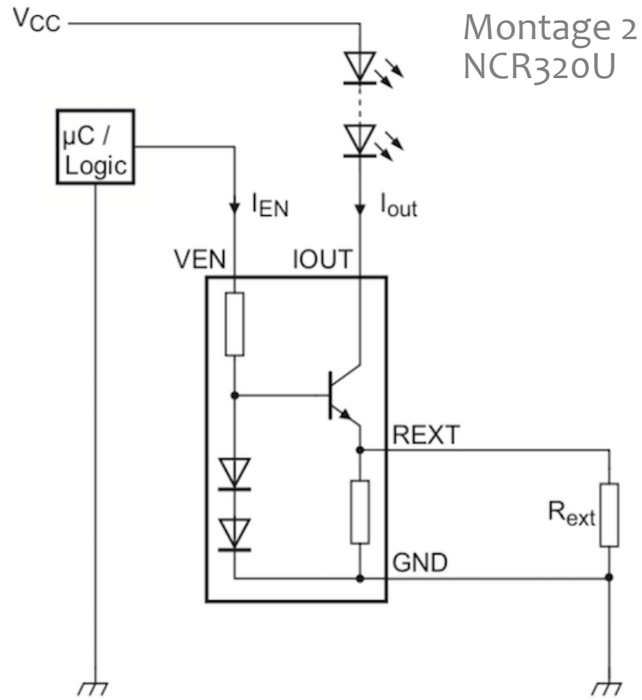


Montage bonus (1 bis)



https://wiki.analog.com/university/courses/electronics/text/chapter-11

Montage 2 NCR320U



N4 / Pilotage d'une barrette CCD (1/2)

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

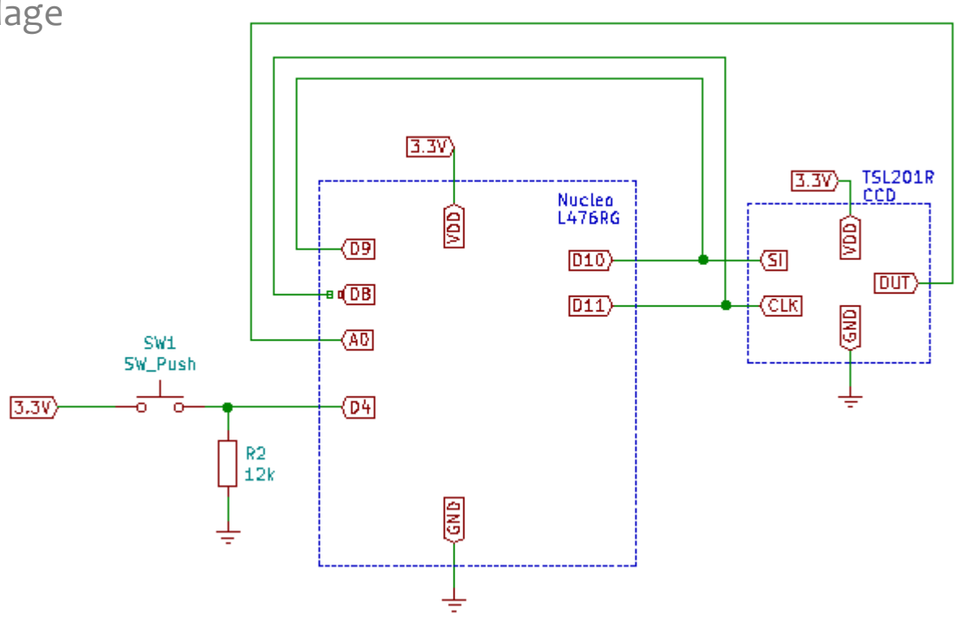
MOTS-CLEFS

+ Microcontroleur
+ Capteur CCD
+ Sortie Série

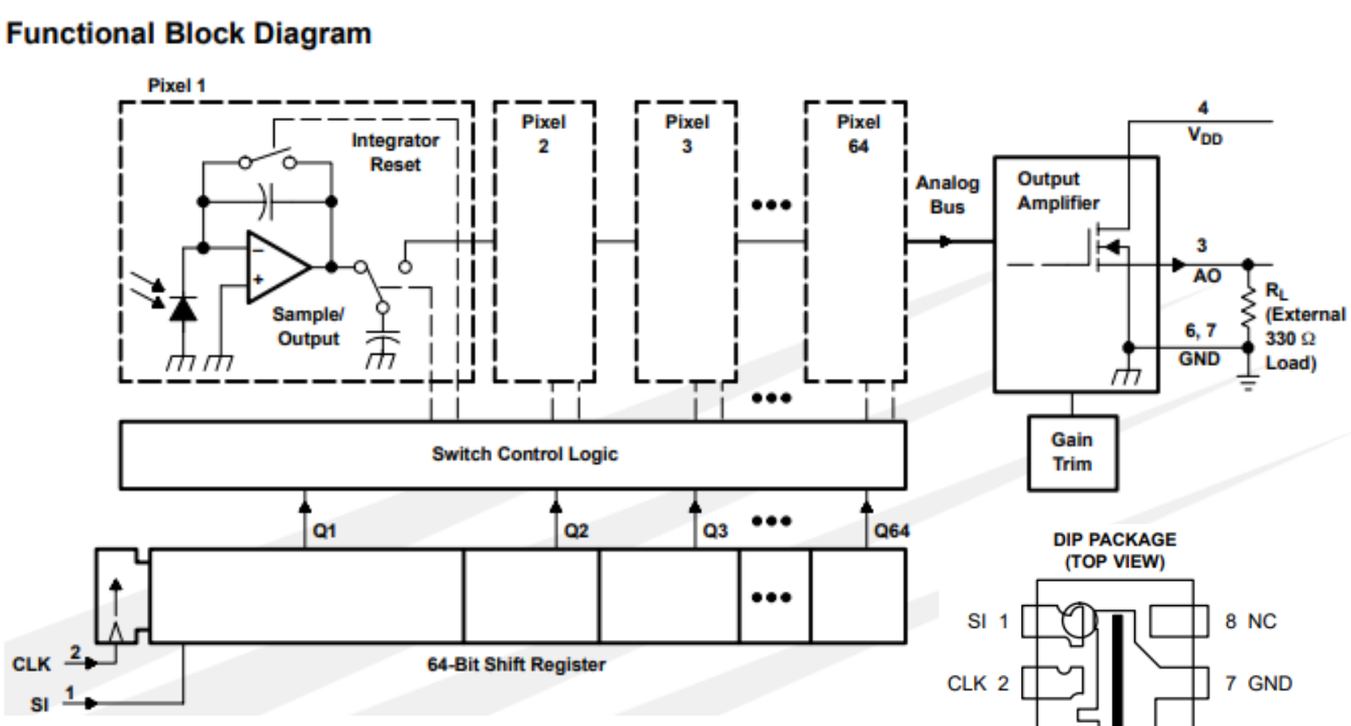
COMPOSANTS

+ Microcontrôleur
+ Barrette CCD - TSL201R

Câblage



Capteur linéaire CCD - TSL201R



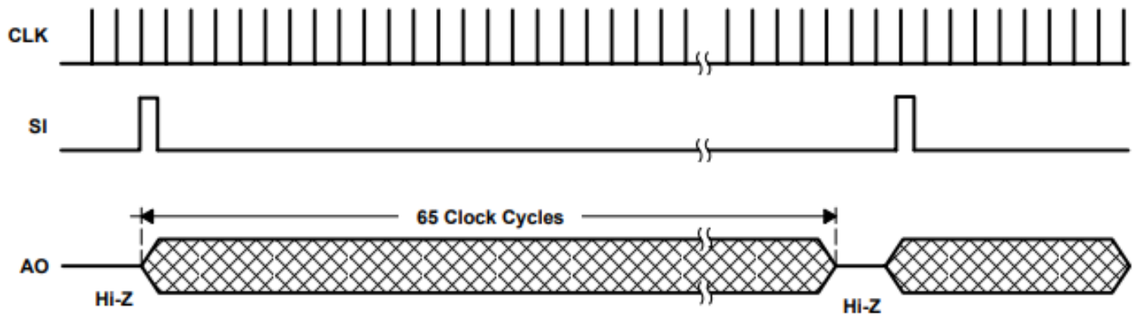
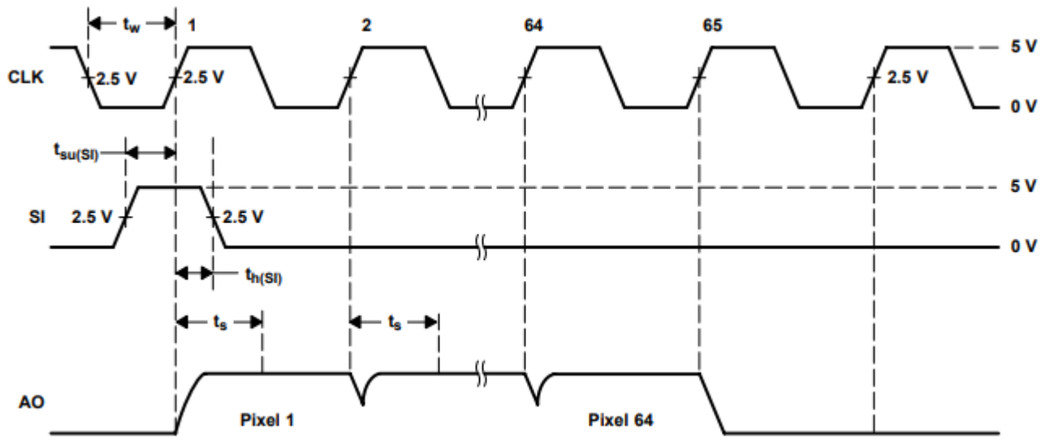


Figure 1. Timing Waveforms



N4 / Pilotage d'une barrette CCD (2/2)

Programme

```
#include "mbed.h"

#define   T_CLK         100
#define   T_SU_SI       20
#define   TOTAL_PER     80

InteruptIn    bp_acquire(D4);
InteruptIn    ccd_CLK_in(D8);

DigitalOut    ccd_SI(D10);
PwmOut        ccd_CLK(D11);
AnalogIn      ccd_out(A0);

int       cpt_acq, acq_OK;
double    data[64];


void   ISR_acquire(void){
   if(cpt_acq < 64){
      data[cpt_acq] = ccd_out.read();
   }
   if(cpt_acq == TOTAL_PER){
      wait_us(T_CLK/2-T_SU_SI);
      ccd_SI = 1;
   }
   cpt_acq++;
}
```

```
void   ISR_reset_SI(void){
   if(ccd_SI.read() == 1){
      ccd_SI = 0 ;
      cpt_acq = 0 ;
   }
}

void   ISR_print_acq(void){ acq_OK = 1 ; }

int main()
{
   ccd_CLK.period_us(T_CLK);
   ccd_CLK.write(0.5);
   ccd_SI = 0;
   acq_OK = 0;
   cpt_acq = 0;

   ccd_CLK_in.fall(&ISR_acquire);
   ccd_CLK_in.rise(&ISR_reset_SI);
   bp_acquire.rise(&ISR_print_aqc);

   while(1){
      if((cpt_acq == 64) && (acq_OK == 1)){
         for(int i = 0 ; i < 64 ; i++)
            printf("d_%d = %lf", i, data[i]);
         acq_OK = 0;
      }
   }
}
```