

1. Capteur de distance infrarouge par réflexion

1.1. Principe

Le principe d'un capteur par réflexion est d'éclairer l'obstacle avec une LED infrarouge, et de mesurer la lumière réfléchie avec une photodiode ou un phototransistor. L'objet éclairé retransmet une énergie inversement proportionnelle au carré de la distance.

Une photodiode, un phototransistor, laissent passer un courant proportionnel à l'énergie lumineuse reçue. Dans l'obscurité totale, il y a un courant de fuite (dark current). Une diode travaille avec des niveaux de courants faibles (surtout les diodes de type PIN), alors qu'un phototransistor génère un courant plus élevé, tout en ayant un courant de fuite proportionnellement plus élevé. La solution d'emploi facile, avec de nombreux composants à disposition, est le phototransistor. Pour une détection de distance dans un petit robot, le Vishay TCRT100 (Spoerle) est une solution économique car la LED et le phototransistor sont dans le même boîtier et ne s'influencent pas. La distance minimale de 1 mm n'est pas un inconvénient.

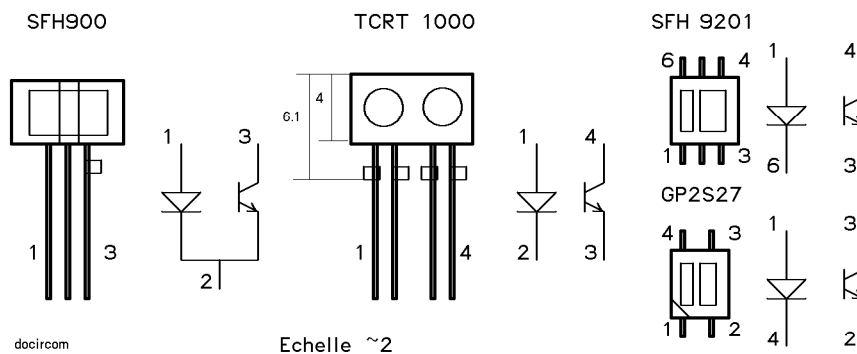


Fig. 1 Capteurs optique par réflexion commerciaux (le SFH900 ne se fabrique plus)

Le schéma de câblage d'un détecteur d'obstacle est évident. Une résistance fixe le courant dans la LED qui éclaire l'obstacle. La résistance du photo-transistor est mesurée soit avec un diviseur de tension, soit avec une conversion courant-temps par mesure du temps de charge ou décharge d'un condensateur. Que choisir, et pourquoi est-il essentiel de commander l'éclairage par un transistor?

1.2. Analyse opto-électronique

Sur un graphique doublement logarithmique, l'intensité réfléchie pour un éclairage donné est une droite de pente -2 , puisque l'énergie reçue est inversement proportionnelle au carré de la distance. La figure 2 concerne le circuit TCRT1000, mais un autre capteur ne donnera qu'une "droite" décalée. Les données du fabricant apparaissent en trait-point. La courbe principale correspond à une LED alimentée en 5V avec une résistance en série de 47 Ohm. Un micro-ampèremètre mesure le courant dans le phototransistor (l'ampèremètre est lié au 5V et la base du phototransistor à la masse).

Le courant avec l'obstacle éloigné dépend fortement de l'éclairage ambiant. Une ampoule à filament génère beaucoup d'infrarouge. Un éclairage néon, ou des ampoules économiques sont nettement préférables. Le "dark current" du capteur est nettement plus faible, et ce paramètre ne joue pas de rôle dans le choix du meilleur capteur: ce qu'il faut c'est minimiser avec des caches l'effet de la lumière ambiante, et savoir que les performances se dégradent lorsque le rapport entre la puissance lumineuse émise par la LED et la lumière ambiante diminue. Par construction, les capteurs comme le TCRT 1000 ont un maximum pour une distance de 1mm environ.

Avec une résistance de 100 Ohm, l'intensité réfléchie est moitié, ce qui réduit la distance mesurable lorsque la lumière ambiante est importante. Une résistance de 22 Ohm ne double pas l'intensité lumineuse, car il y a saturation et échauffement dans la LED. D'où l'avantage d'utiliser une diode de puissance et un photo-transistor séparés pour mesurer de plus grande distance. Mais cela coûte de l'énergie électrique, des caches, de l'optique.

Pour faire au mieux avec notre capteur TCRT1000, il faut mesurer la lumière ambiante, avec la LED éteinte, pour savoir quel est le domaine de validité des mesures. On peut soustraire le courant dû à la lumière ambiante au courant éclairé, mais dans la gamme où la mesure est raisonnablement précise, cela ne fait pas grande différence, même si l'échelle logarithmique donne l'impression que oui.

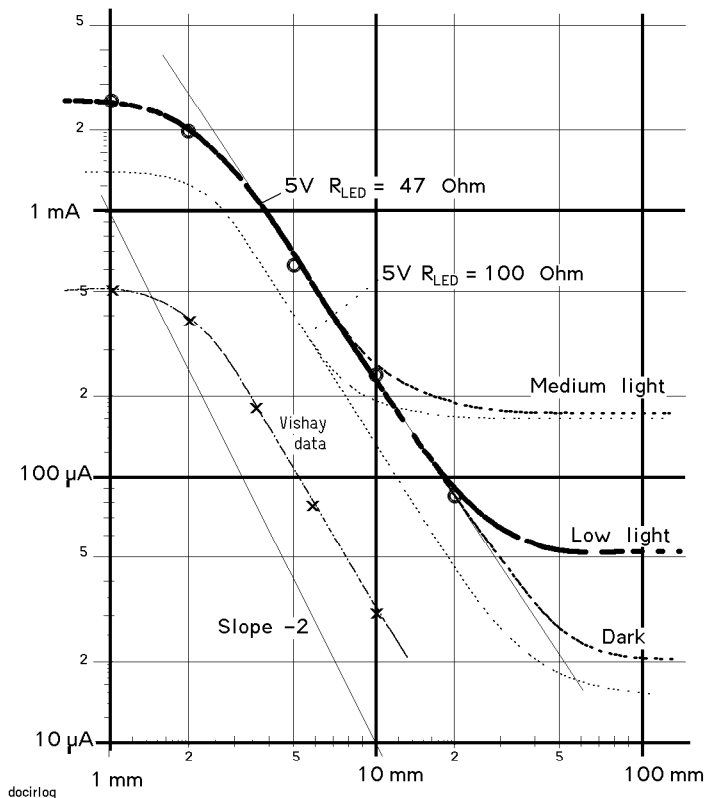


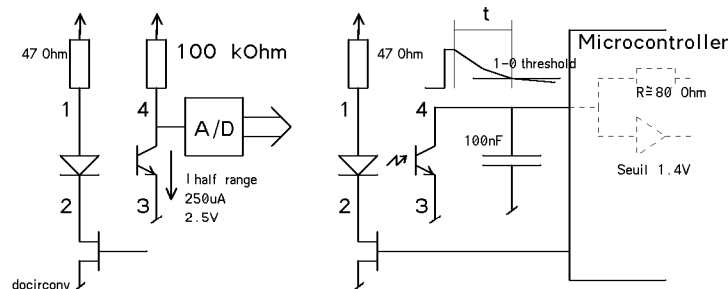
Fig. 2 Courant dans le photo-transistor en fonction de la distance

A partir de cette analyse, l'algorithme de mesure est assez évident. Avec un microcontrôleur, il ne faut pas oublier que l'on travaille efficacement en 8 bits, et qu'il est exclu de penser à de la virgule flottante, comme notre échelle logarithmique nous y encourage. Dans l'obscurité, la mesure couvre 2 ordres de grandeurs. Il serait absurde de chercher à donner au robot une mesure de distance précise, même à 10%, entre 1mm et 100mm. Ce qui compte est la monotonicité; si la distance augmente, la mesure doit varier dans le même sens. Pour le robot, les distances sont aussi de type logarithmique, et une dizaine de valeurs (codage sur 4 bits) peuvent être bien suffisantes pour un comportement cohérent. Si l'on passe essentiellement par des tables pour faire les calculs, ce qu'impose un microcontrôleur, c'est le plus simple et efficace. La monotonicité est plus importante à gérer lorsque le codage est grossier.

L'algorithme revient donc à mesurer d'abord le courant obscur. Si la valeur éclairée est le double on convertit cette valeur en distance selon une loi quadratique (en passant par une table). En dessous, une table prenant en compte la valeur éclairée et la différence (ou les deux valeurs) peut approximer la distance. Pour la mise au point, et la vérification en tout temps du bon fonctionnement, un affichage approprié doit pouvoir être mis en service.

La mesure du courant avec un diviseur de tension et un convertisseur A/D n'est pas intéressante, car la linéarité de la conversion n'est bonne que lorsque les valeurs des résistances sont proches. On pourrait être tenté d'utiliser un convertisseur 12 bits pour mieux exploiter les extrémités de gammes, mais il y a le bruit et la précision relative à maîtriser. Avec un microcontrôleur, il est facile de mesurer un temps, et ce temps peut couvrir plusieurs ordres de grandeur dans une application robotique. 100 microsecondes est mesurable par le microcontrôleur, 100 millisecondes (3 ordres de grandeur plus lent) est encore un temps très court pour un robot.

Le convertisseur courant-temps est particulièrement facile avec une ligne bidirectionnelle d'un microcontrôleur. Le micro impose l'état "1" (tension 5V), attend que le condensateur soit chargé, commute en entrée et mesure le temps de décharge jusqu'à ce que la tension passe en dessous du seuil de transition état "1" vers état "0". La précision en tension et en température n'est pas excellente, mais ces deux paramètres sont assez constants pour un robot en chambre, et une précision de 10% avec un système simple et fiable nous comble.



docirconv

Fig. 3 Conversion de l'intensité lumineuse en information numérique

La mesure du temps se fait avec un timer ou dans une boucle d'attente. La mesure non éclairée donne des valeurs dont on connaît l'ordre de grandeur. Pour la mesure éclairée, la décharge peut être très rapide à courte distance. Une commutation automatique de gamme, en changeant la période du timer ou en éclairant par impulsions, est envisageable si on ne veut pas travailler avec des compteurs 16 bits.

1.3. Autres capteurs de lumière

Pour des performances supérieures, tout est dans l'optique et la qualité de réflexion des obstacles. A la place d'un phototransistor, on peut utiliser un capteur "Light to voltage" de type TSL253/254 ou TSL256 (anciennement Texas Instrument, actuellement TAOS, certains types ne sont plus fabriqués). Un ampli incorporé donne une tension proportionnelle à l'énergie lumineuse reçue, ce qui ne nous intéresse pas, car il faut ajuster le domaine de fonctionnement en changeant de circuit et en agissant sur l'optique. Les circuits "Light to Frequency" TSL245 TSL220 et TSL235 sont par contre très intéressants car ils mesurent la lumière sur plusieurs décades. Pour le microprocesseur, mesurer une fréquence est assez facile; pour une fréquence élevée (10 kHz et plus), on mesure la fréquence en comptant les impulsions pendant 10 ms par exemple. Pour une fréquence basse, on mesure une période avec un timer incrémenté par l'horloge du processeur.

La modulation permet de mieux repérer un signal modulé dans la lumière ambiante. Ceci convient bien pour un capteur avec un seuil en tout-ou-rien (voir le document décrivant les circuits IS471F et S4282). Pour avoir une mesure proportionnelle, il faudrait une électronique délicate avec des amplificateurs logarithmiques.

Les capteurs par triangulation utilisent un autre principe. Voir sur le site Didel le document décrivant le capteur Sharp GP2D02L.

Conversion de nombres entiers en code logarithmique

Le problème précis est de faire correspondre à la fréquence (ou période) d'un TSL230/235 une valeur logarithmique 8 bits. Le TSL couvre 5 décades, limitons-nous à $65356 = 2^{16}$ pour limiter le comptage à 16 bits, et la période de mesure pour qu'elle soit inférieure à 0.1 seconde (fréquence max 2 MHz).

La correspondance à établir entre la valeur linéaire 16 bits et son code logarithmique 8 bits est

Nombre d'impulsions (période)	Valeur en code logarithmique
1	0
2	16
4	32
8	48
...	
1024	160 = 10 x 16 car 1024 = 2 ¹⁰
...	
2 ¹⁶	256

La dernière ligne du tableau est en fait la borne supérieure non comprise du domaine de mesure. La formule de conversion fait correspondre à $X = 2^n$ la valeur $n \times 16$:

$$n = \log_2 X = \log_{10} X / \log_{10} 2$$

$$(\log_{10} 2 = 0.30103)$$

Pour interpoler avec une précision de $1/16 = 6\%$, il suffit de regarder les 4 bits qui suivent le poids forts, et de passer par une table pour avoir la valeur à additionner. Pour plus de précision, on peut agrandir la table à 32 (3%) ou à 64 valeurs (1,5%).

0000		0
0001	$(\log 17/16 / 0.30103) * 16$	1,40 ≈ 1
0010	$(\log 18/16 / 0.30103) * 16$	2.72 ≈ 3
...		
1000	$(\log 24/16 / 0.30103) * 16$	≈ 10
...		
1111	$(\log 31/16 / 0.30103) * 16$	15,25 ≈ 15

Si on veut la fréquence au lieu de la période, il suffit de complémentier le résultat.

Le programme résultant est donc très simple. On décale en comptant jusqu'à ce qu'il n'y ait plus de zéros non significatifs. On prend ensuite les 4 ou 6 bits suivants pour passer par une table pour obtenir la valeur logarithmique à ajouter.

JDN 26.11.00 22.04.04

L'algorithme de conversion regarde le bit de poids fort du mot de 16 bits. S'il vaut 1, la valeur est $r \times 16$ (r =rang), plus la contribution des bits suivants.

1xxxxxxx	yyyyyyyy	≤32k	Fx
01xxxxxx	yyyyyyyy	≤16k	Ex
001xxxxx	yyyyyyyy	≤8k	Dx
0001xxxx	yyyyyyyy	≤4k	Cx
00001xxx	yyyyyyyy	≤2k	Bx
000001xx	xyyyyyyy	≤1k	Ax
0000001x	xxxxyyyy	≤512	9x
00000001	xxxxyyyy	≤256	8x
00000000	1xxxxxxx	≤128	7x
00000000	01xxxxxx	≤64	6x
00000000	001xxxxx	≤32	5x
00000000	0001xxxx	≤16	4x
00000000	00001xxx	≤8	30 32.. 3E
00000000	000001xx	≤4	20 24.. 2C
00000000	0000001x	≤2	10 18
00000000	00000001	≤1	00
00000000	00000000	!	-∞