

BasicBot

version 1.0

Robot

Mobile

Programmable

Le BasicBot est un robot mobile programmable qui exploite le microcontrôleur "PICBASIC-3B" du fabricant Comfile Technology. Sa programmation très simple en langage BASIC vous permettra de gérer ses 2 moteurs ainsi que l'ensemble de ses capteurs, lesquels offrent une multitude de combinaisons et possibilités de fonctionnement possibles.

- Description sommaire
 - : Moteurs CC, Capteurs IR, Buzzer , LED, Alim.: 9V
 - : Dim.: 80 x 94 × 60 mm
- Géré par un PICBASIC-3B programmable en langage BASIC
- Idéal pour l'initiation, l'enseignement....
- Simple à mettre en oeuvre.
- Nombreux types de fonctionnements possibles

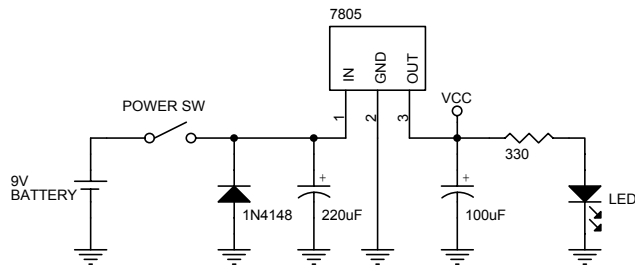
Attention ce robot n'est pas un jouet, veuillez le laisser impérativement hors portée des enfants de moins de 10 ans, qui pourraient avaler certaines petites pièces. A ce titre, Lextronic n'est pas responsable des dommages quelques soient pouvant résulter d'une mauvaise utilisation de ce robot.

Programmable
Mobile Robot
BasicBot

ETUDE ET DESCRIPTION DU SCHEMA THEORIQUE DU BASIC-BOT

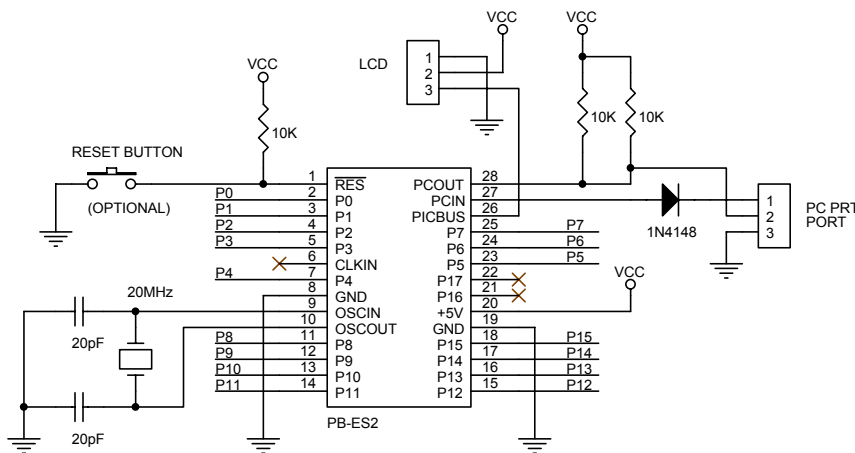
Génération de la source 5 V

Un régulateur standard (7805) est utilisé pour convertir la tension de 9 V en 5 V. Une diode (1N4148) à l'entrée sert de protection contre les mauvaises connexions. Lorsque le 5V est généré, la Led doit s'allumer pour signaler un état normal.



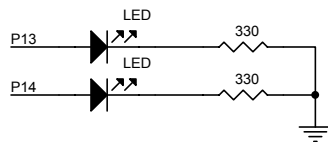
Le cœur du système

Le BASIC-BOT est conçu à partir d'un circuit intégré «PICBASIC-3B» (ici référencé PB-ES2). Ce dernier est en fait un microcontrôleur spécialement programmé pour pouvoir interpréter et exécuter des programmes en langage BASIC. Ces programmes pourront être écrits sur un compatible PC, puis transférés dans la mémoire Flash du PICBASIC-3B à l'aide d'un câble qu'il faudra raccorder au port parallèle du PC (consultez la notice jointe des PICBASIC-3B pour plus d'informations). Le schéma ci-dessous reprend la mise en œuvre «type» du PICBASIC-3B (à noter toutefois que le bouton RESET ainsi que les ports P16, P17 et CLKIN ne sont pas utilisés sur le robot).



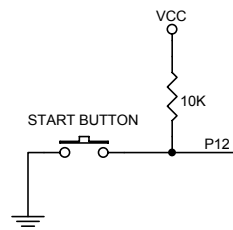
Les Leds d'indications de direction

Ces 2 Leds sont connectées à P13 et P14. P13 pilote la Led Gauche et P14 celle de droite. Un niveau logique «haut» sur ces sorties fait allumer les Leds associées (un niveau logique «bas» les éteints).



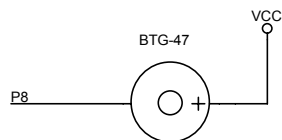
Le bouton-poussoir

Ce dernier (utilisé comme bouton de «démarrage» du programme) est relié au port P12. Lorsqu'il n'est pas sollicité, P12 reçoit un niveau logique «haut», puis « bas » lorsque l'on appui sur le bouton-poussoir.



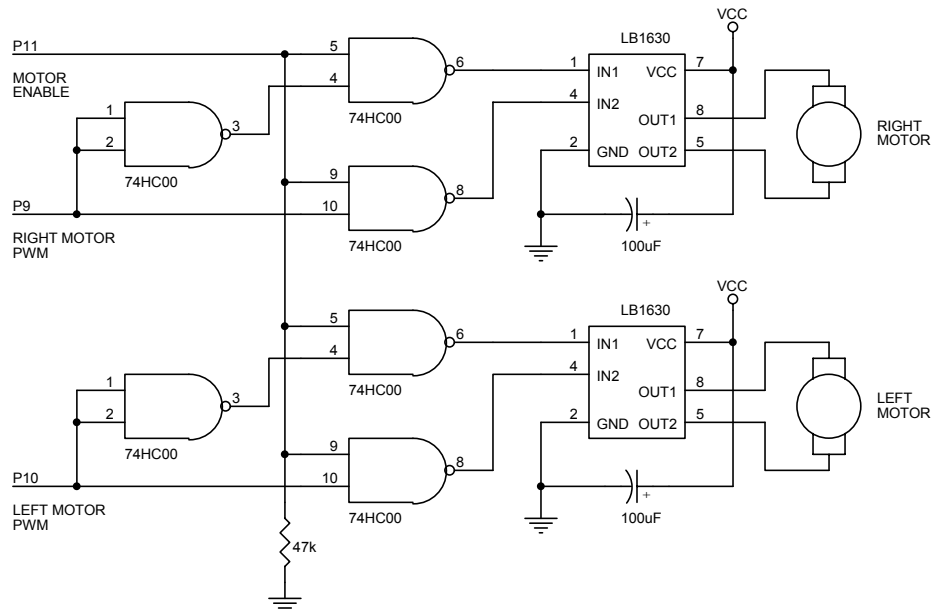
Le buzzer

Ce dernier est relié au port P8. Sans oscillateur intégré, il nécessitera qu'on lui envoi un signal carré pour générer des sons.



La commande des moteurs

Celle-ci repose sur l'utilisation de 2 circuits intégrés spécialisés «LB1630».



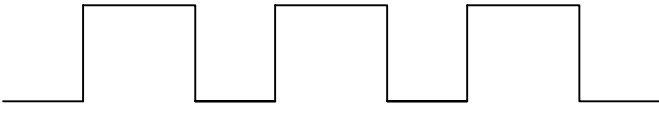
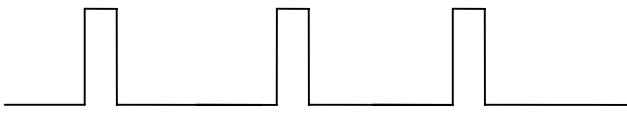

Ces derniers, associés à une logique de commande, vous permettront de piloter entièrement le robot selon les données du tableau ci-dessous. Les condensateurs de 100uF servent à éliminer les parasites générés par les moteurs lors de leur rotation.

IN1	IN2	OUT1	OUT2	Etat du moteur
1	0	1	0	Avant
0	1	0	1	inversion
1	1	off	off	Arrêt
0	0	off	off	Arrêt

Les entrées «IN2» sont pilotées par le port «P11» du PICBASIC (via une circuiterie logique) qui validera ou non le pilotage des moteurs.

Les entrées «IN1» de chaque moteur sont pilotés par les ports «P9» et «P10» du PICBASIC. Ces ports sont spécialement conçus pour pouvoir générer des signaux PWM en tâche de fond (c'est à dire en même temps que l'exécution du reste de votre programme).

Ce type de signal est spécialement adapté au pilotage des moteurs à courant continu comme ceux qui équipent le BASIC-BOT. Ainsi le rapport cyclique du signal sera directement proportionnel à la vitesse du moteur (voir graphes ci-dessous).

Rapport	Signal
D=50%	
D=25%	
D=75%	

A noter toutefois que les circuits intégrés «LB1630» disposent d'un mode de fonctionnement différent vis-à-vis des signaux «PWM». Ce mode permet ainsi de gérer très simplement le sens de rotation des moteurs. Lorsque le circuit reçoit un signal de rapport cyclique de 50 %, il considèrera que le moteur associé devra rester immobile.

Si le rapport cyclique est supérieur à 50 %, il fera tourner le moteur dans un sens (la vitesse étant plus ou moins importante au fur et à mesure que le rapport cyclique du signal approche des 100%).

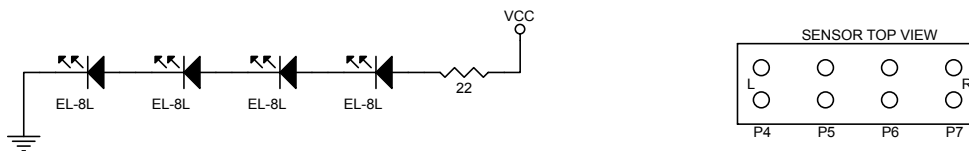
Si le rapport cyclique est inférieur à 50 %, il fera tourner le moteur dans l'autre sens (la vitesse étant plus ou moins importante au fur et à mesure que le rapport cyclique du signal approche des 0 %).

Les capteurs infrarouges au sol (détection numérique)

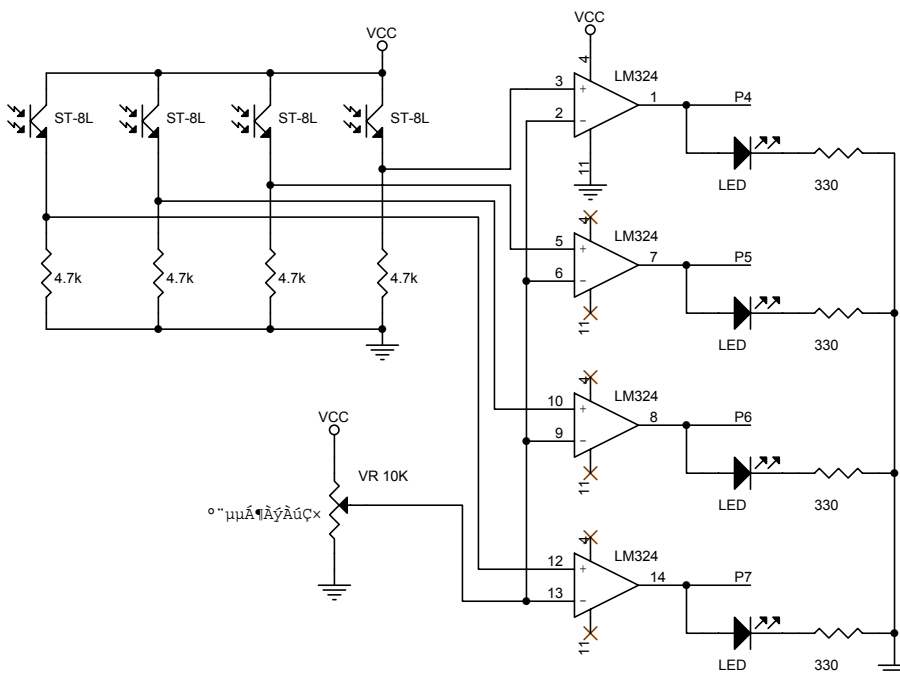
Le BASIC-BOT est équipé de 4 paires de détecteurs infrarouges qui lui permettront de détecter une ligne noire sur un sol «clair» (idéal pour réaliser un robot suiveur de ligne).



Ces détecteurs sont composés de Leds infrarouges émission montées en série et constamment alimentées via une résistance de 22 ohms.

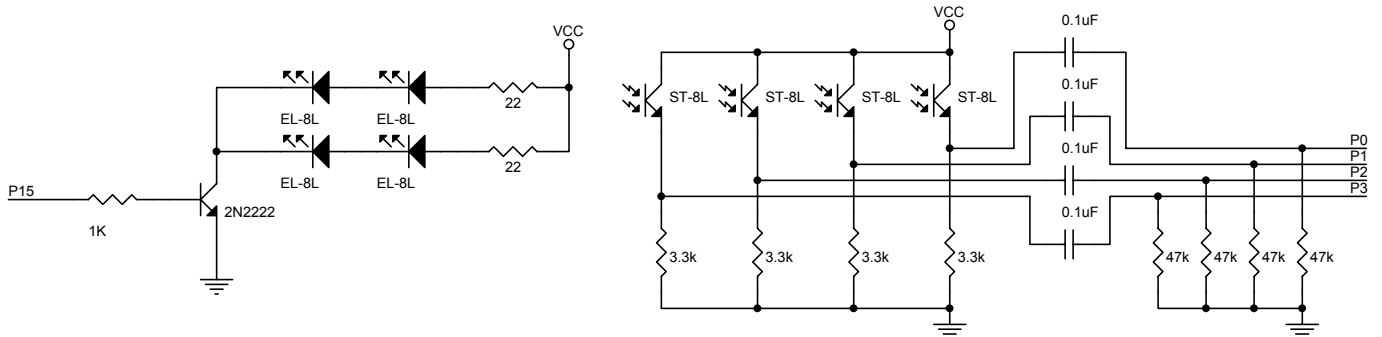


A chaque Led émettrice, est associée un photo-coupleur dont la sortie est appliquée à l'entrée d'un amplificateur opérationnel (monté en comparateur). En présence d'une surface claire, le faisceau IR sera réfléchi (alors qu'une surface foncée (comme un trait noir) ne réfléchira pas l'émission IR). Il en résulte que la tension en sortie des phototransistors variera en fonction de la couleur du sol présent sous le robot. Un potentiomètre permet le réglage d'une tension de seuil à partir duquel les sorties des amplificateurs opérationnels «basculent» ou non au niveau «haut». Dans le cadre d'un réglage «normal», les ports P4, P5, P6 et P7 recevront un niveau logique «haut» lorsque le robot sera placé sur une surface claire (les Leds associées à chaque sortie des amplificateurs opérationnels s'allumeront alors). A l'inverse, lorsqu'il est placé sur un sol foncé, les Leds s'éteindront et les ports P4, P5, P6 et P7 recevront un niveau logique «bas». Chaque sortie étant indépendante, il sera alors possible (en fonction de celle présentant ou non un niveau logique « haut ») de savoir où se trouve une ligne noire par rapport au centre du robot). A noter que le fonctionnement des Leds de signalisation est permanent et indépendant de celui du PICBASIC-3B (Ainsi même le BASIC-BOT non programmé, celle-ci s'allumeront et s'éteindront en fonction de la nature du sol présent sous le robot). La figure en haut à droite montre la répartition des ports P4 à P7 en fonction de l'emplacement des capteurs.



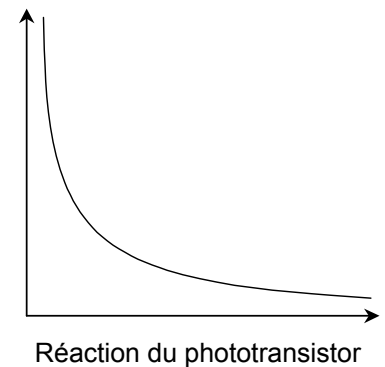
Les capteurs infrarouges horizontaux (détection analogique)

Le BASIC-BOT est équipé de 4 paires de détecteurs infrarouges placés devant et sur les côtés, lesquels lui permettront de détecter un obstacle devant lui.



Ces détecteurs sont composés de Leds infrarouges émission pilotées par un transistor 2N2222, lui-même activé par le port P15 du PICBASIC-3B.

4 Phototransistors associés à ces Leds infrarouges sont directement reliés aux ports P0 à P3 du microcontrôleur. De part la variation de la tension de sortie de ces deniers en fonction du niveau de réception infrarouge, il sera assez simple de déterminer la présence ainsi qu'une estimation de la portée d'un obstacle se trouvant devant chacun s'entre eux. Pour ce faire, le programme du PICBASIC-3B alimentera les Leds infrarouges pendant un bref instant pendant lequel il viendra «lire» la valeur analogique d'un des ports, puis il recommencera pour chacun des autres ports. Pour rappel, les ports P0 à P3 peuvent être configurés en sortie, en entrée « tout-ou-rien » ou en entrée capable de lire une valeur analogique comprise entre 0 et 5 V en vous restituant un nombre proportionnel compris entre 0 et 1023 (conversion numérique sur 10 bits).



Répartition/attribution des ports et capteurs IR

Récapitulatif de l'allocation des ports du microcontrôleur vis-à-vis des fonctions du robot

Les Leds, capteurs, moteur, buzzer sont affectés comme suit aux ports du PICBASIC-3B:

Composant	Port No.	Fonction
Capteur (reflection sur côté)	P0 - P3	Détection d'un obstacle et de sa distance sur les côtés et devant. P0, P1, P2, P3 en partant de la gauche. Plus l'obstacle est proche, plus la tension est élevée. Sortie: 0 - 5V
Capteur (reflection au sol)	P4 - P7	Détection de l'existence de lignes sur le sol. P4, P5, P6, P7 P3 en partant de la gauche. La reflection depuis le sol génère un niveau "haut" (sinon "bas") D1 - D4 : affiche l'état des capteurs.
Leds	P13	Passe à l'état "haut" lorsque la sortie de la Led gauche est "haute".
	P14	Passe à l'état "haut" lorsque la sortie de la Led droite est "haute".
Bouton-poussoir	P12	Bouton-poussoir de "départ" Génère un niveau logique "bas" sur sollicitation
Buzzer piezo	P8	Génération des sons (la tonalité est fonction de la fréquence appliquée).
Moteurs	P11	Signal "M/A" des moteurs. Lorsque P11 est à l'état "haut", les moteurs peuvent fonctionner. A l'état "bas", ces derniers ne pourront plus tourner. Lorsque P11=1, la vitesse des moteurs est alors gérée par les signaux P9, P10.
	P9	Génère un signal PWM déterminant la vitesse du moteur droit. D=50% Stop, D>50% marche avant, D<50% marche arrière Plus le rapport est supérieur à 50%, plus la vitesse est élevée.
	P10	Génère un signal PWM déterminant la vitesse du moteur gauche. D=50% Stop, D>50% marche avant, D<50% marche arrière Plus le rapport est supérieur à 50%, plus la vitesse est élevée.

Liste des composants du BASIC-BOT

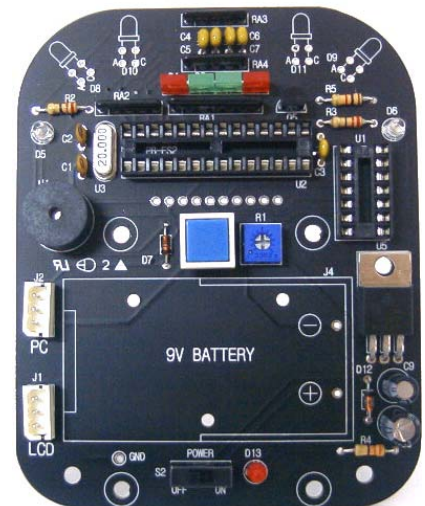
Composant	Spécifications	Qté	Repère
Module PICBASIC-3B	PB-ES2	1	U2
Driver pour moteur	LB1630	2	U7,U9
Régulateur	LM7805	1	U5
IC	74HC00	2	U6,U8
	LM324	1	U1
Quartz	20MHz ATS	1	U3
Condensateurs	Electrolytic 100uF/16V	3	C9,C12,C13
	Electrolytic 220uF/16V	1	C8
	Mono 0.1uF (104)	7	C3 - C7,C10,C11
	Céramique 20pF	2	C1,C2
Bouton-poussoir	Bouton-poussoir	1	S1
Inter «M/A»	Modèle miniature	1	S2
Connecteurs	MOLEX 5267-3	2	J1, J2
	MOLEX 5267-2	2	J6, J7
	1 ligne de 10 points	2	J3, J5
Leds	Translucides 3mm (Transparent)	2	D5, D6
	Standard 3mm (Rouge)	1	D13
	Rectangulaire (Rouge)	2	D1, D4
	Rectangulaire (Verte)	2	D2, D3
Leds IR	EL-8L (Transparente)	8	D8 - D11, D14 - D17
Phototransistors	ST-8L (Noir)	8	Q1 - Q4, Q6 - Q9
Diode	1N4148	2	D7, D12
Transistor	2N2222	1	Q5
Résistances	22Ω	3	R2, R3, R6
	1KΩ	1	R5
	47KΩ	1	R7
	330Ω	1	R4
Réseaux de résistances	7pin 331	1	RA1
	5pin 103	1	RA2
	5pin 332	1	RA3
	5pin 473	1	RA4
	5pin 472	1	RA5
Résistance variable	103 (10K)	1	R1
Buzzer piezo	BTG-47	1	U4
Support de CI	8P DIP	2	U7, U9
	14P DIP	3	U1, U6, U8
	28P DIP	1	U2
Support de pile 9 V	9V	1	J4
Connecteur	22mm 10pin	1	

Composant	Spécifications	Qté	Repère
Bloc moteur	Intégrant les pneus	1	
Coussinets	Avant/arrière	2	
Vis	M2×6 (Round head)	3	
	M3×5 (Round head)	10	
Ecrous	M2	3	
Entretoise	10mm M	4	
	15mm F	4	
Circuits imprimés	Partie haute et basse	2	
Câble programmation	Pour PICBASIC-3B	1	

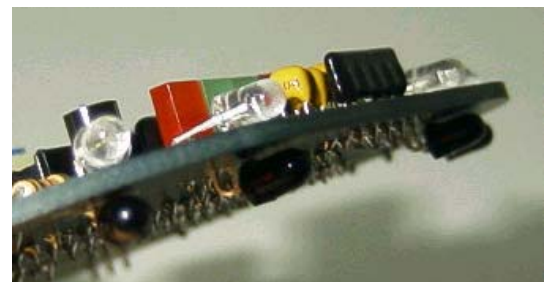
MONTAGE DU «BASIC-BOT»

La réalisation de ce kit est destinée aux personnes de plus de 14 ans. Elle nécessite un maximum d'attention et le plus grand soin aux soudures et au montage. Le circuit imprimé professionnel de type double face avec trous métallisés ne permet pas en effet, en cas d'erreur de montage de dessouder facilement le composant sans abîmer les pistes et pastilles, il faut donc prendre le plus grand soin pour réaliser les soudures sans trop les surchauffer, afin de ne pas détériorer le circuit imprimé ce qui rendrait dans certain cas le dépannage impossible.

1) Montez en premier lieu le circuit imprimé correspondant à la partie haute du robot. Respectez le sens des supports de circuits intégrés (encoche de repérage), ainsi que le sens des réseaux de résistances (carré de repérage associé au point du réseau), ainsi que celui des connecteurs (voir sérigraphie), ainsi que celui des leds (l'anode «A» est matérialisée par la patte la plus longue), ainsi que celui des condensateurs et des diodes (voir sérigraphie).



2) Montez ensuite les phototransistors (couleur noire) puis les Leds IR (translucides). **Respectez cet ordre** pour faciliter les soudures et ne pas vous tromper dans l'implantation. Avant l'opération de soudure pliez les pattes de ces derniers à 90° comme représenté sur le schéma ci-contre et positionnez les sur le circuit (le léger méplat doit se trouver vers l'extérieur du montage, la partie sans méplat se trouvant ainsi contre le circuit imprimé – ceci est valable pour les phototransistors ainsi que pour les Leds IR). Dans tous les cas, les leds IR et les phototransistors ne doivent pas dépasser du circuit imprimé et doivent rester très légèrement en retrait comme indiqué sur la sérigraphie).



3) Assemblez ensuite le circuit imprimé correspondant à la partie basse du robot en respectant les mêmes vérifications de polarité indiquées ci-dessus sur les supports de circuits intégrés, les condensateurs, réseaux de résistances, etc, etc...

4) Repliez à 90° les pattes des 4 leds IR transparentes D14, D15, D16 et D17 en respectant le branchement (l'anode étant la patte la plus longue). Positionner et soudez ces 4 leds en ayant soin de faire passer la partie sphérique au maximum au travers du trou prévu, afin que l'émission infrarouge soit dirigée "vers le sol".

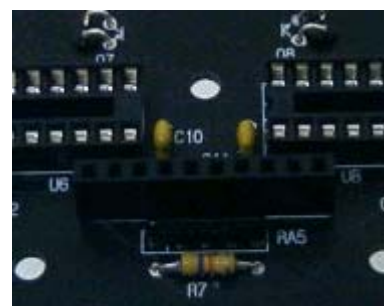
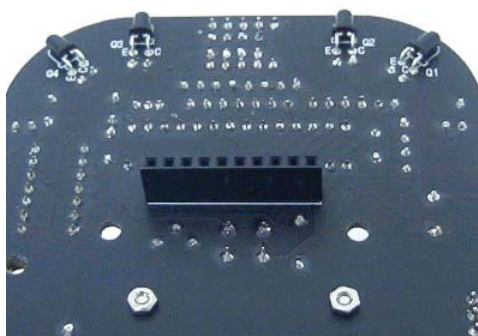
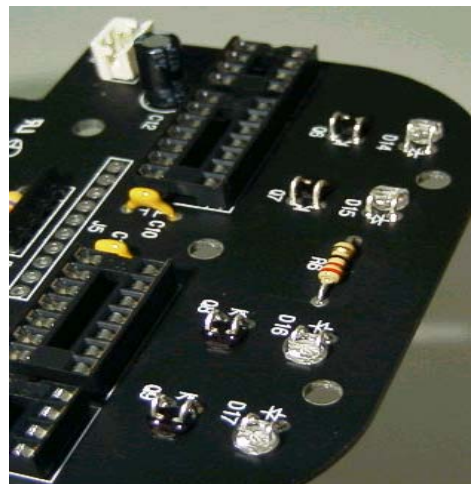
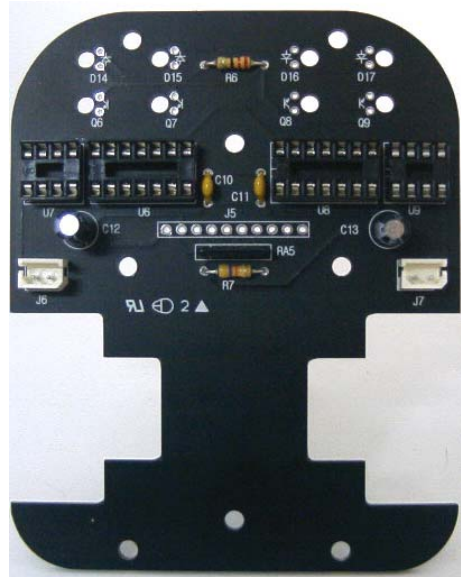
Replier à 90° les pattes des 4 phototransistors (couleur noire) Q6, Q7, Q8 et Q9 en respectant le branchement (l'émetteur correspondant à la patte la plus longue et le collecteur à la plus courte). Positionner et souder ces 4 leds en ayant soin de faire passer la partie sphérique au maximum, au travers du trou prévu et dirigées "vers le sol" comme pour les leds précédentes.

5) Fixer en dessous du circuit imprimé les 2 embouts demi-sphérique en plastique noir à l'aide des vis cruciformes.

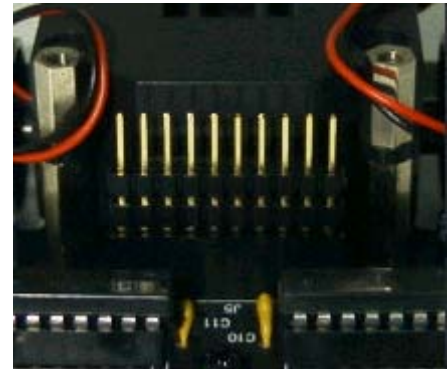
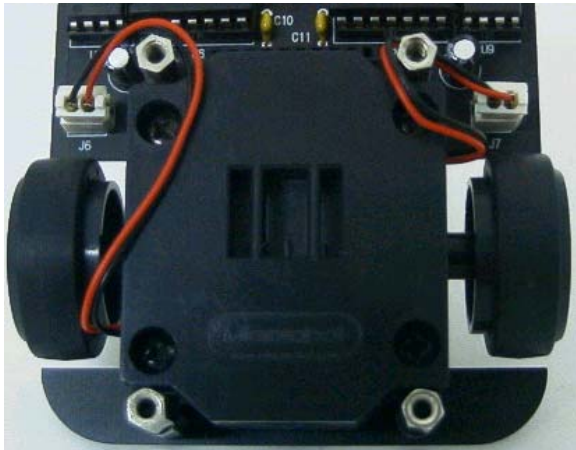
6) Soudez le connecteur femelle côté soudure du circuit imprimé correspondant à la partie haute du robot. Soudez ensuite (côté composant) le connecteur sur le circuit imprimé correspondant à la partie basse du robot.

7) Insérez les circuits intégrés sur leur support en respectant scrupuleusement le sens de leur implantation.

8) Soudez et montez le reste des composants.



- 9) Fixez les entretoises sur le bloc moteur. **A noter qu'il est impératif de ne pas tourner les roues manuellement sans quoi vous risquez de casser les engrenages plastiques du bloc moteur (sans échange gratuit de notre part)** – En cas de «casse», il vous est possible d'acheter au détail le bloc moteur seul (consultez-nous).
- 10) Raccordez les 2 platines via les supports prévus à cet effet et vérifiez une nouvelle fois l'intégralité du montage (sens des composants, valeurs, etc...).



- 11) Procédez enfin au montage final de la pile (non livrée). Attention à sa polarité.

Note importante : Il est impératif que vous utilisiez une pile de type **alcaline** afin d'assurer un fonctionnement « normale » à votre BASIC-BOT. Toutefois, il faut savoir que ce dernier consomme près de 280 mA et qu'en condition normale la pile sera amenée à être très vite utilisée. Il sera donc utile dans le cadre d'une mise au point prolongée (dans les milieux éducatifs par exemple) d'utiliser une source alimentation externe (type alim. de labo) et de mettre la pile sur le robot une fois le programme achevé.



Installation du logiciel vous permettant de programmer le robot

Afin de pouvoir programmer votre Basic-Bot, il vous faudra installer le logiciel "PICBASIC-LAB" sur votre PC (consultez la notice des PICBASIC pour davantage d'informations). Ce logiciel se trouve sur le CD-ROM joint avec le robot. En cas de problème, consultez la FAQ dédiée aux PICBASIC sur notre site internet www.lextronic.fr

Programmation du robot

Afin de pouvoir programmer votre Basic-Bot, vous devrez raccorder ce dernier au port parallèle de votre PC à l'aide du câble joint (si vous disposez déjà d'un PICBASIC, il vous faudra utiliser le câble du robot dont le connecteur en bout diffère de celui des PICBASIC «standards»). Le connecteur marqué «PC» est dédié à cet usage. Celui marqué « LCD » permettra la connexion d'un afficheur LCD à commande série Comfile technology (modèle 2 x 16 caractères par exemple de référence ELCD162). On veillera à ne pas se tromper de connecteur sous peine de destruction potentielle. Le robot devra bien évidemment être mis sous tension pour pouvoir être programmé. Il est très fortement conseillé (comme dans les exemples ci-dessous) d'éviter de faire fonctionner directement les moteurs dès la mise sous tension (afin d'éviter la génération de parasites lors de la programmation). Il est ainsi conseillé de ne procéder à la commande des moteurs qu'après que l'utilisateur est sollicité le bouton-poussoir du robot.

EXEMPLES DE PROGRAMMES

ROBOT "LINE TRACER"

Le programme ci-dessous est chargé d'origine dans le PICBASIC-3B du BasicBot (avec des valeurs de vitesse différentes). Il s'agit d'un exemple qui montre comment le robot peut être configuré pour suivre une ligne noire tracée sur le sol. Sachant que la "réflectivité" de la couleur noire est faible et que dans ce cas le capteur vous retournera un niveau logique "bas" et qu'à l'inverse la couleur blanche générera un niveau logique "haut", il en résulte que si P5 et P6 retournent un niveau "bas" (0) et "haut" (1), on peut en déduire que le côté gauche du robot est sur une ligne. Alors le programme réduira la rotation du moteur gauche. Lorsque P5=1 et P6=0, le côté droit du robot est sur une ligne. Alors le programme réduira la rotation du moteur droit. Avec ce procédé, le robot sera alors en mesure de suivre une ligne noire tracée au sol.

Le robot démarre son fonctionnement avec un signal sonore juste après que vous aurez pressé son bouton-poussoir. Ce dernier devra au préalable avoir été posé sur un parcours fabriqué à partir d'une ligne noire d'épaisseur convenable (2 cm env.) sur fond blanc formant un circuit fermé aux courbures **très faibles**. La résistance ajustable R1 sert à modifier la sensibilité de détection des capteurs au sol. A l'état normal, ces derniers doivent allumer les Leds D1 à D4 sur un fond blanc et les éteindre sur un fond noir.

```
REM [ DESCRIPTION DU PROGRAMME DISPO SUR LA DISQUETTE JOINTE] -----
' PROGRAMME "LINE TRACER"

REM [ DECLARATION DES VARIABLES ]-----
DIM I AS BYTE
DIM J AS INTEGER
DIM S0 AS BYTE
DIM S1 AS BYTE

REM [ DECLARATION DES CONSTANTES ]-----

CONST R_MOTOR = 9
CONST L_MOTOR = 10
CONST MOT_ON = 11
CONST PIEZO_BUZZER = 8
CONST START_BUTTON = 12

CONST STOP = 128
CONST SPEED_NORMAL = 170
CONST SPEED_DOWN = 120

CONST SENSE_LEFT_OUT = 4
CONST SENSE_LEFT_IN = 5
CONST SENSE_RIGHT_IN = 6
CONST SENSE_RIGHT_OUT = 7

CONST LED_LEFT = 13
CONST LED_RIGHT = 14

REM [ MAIN ]-----

POKE &H12,&H34 ' CHANGEMENT DU PREDIVISEUR DU SIGNAL PWM (1:16 > 1:1)

BEEP PIEZO_BUZZER
PWM R_MOTOR,STOP
PWM L_MOTOR,STOP
OUT MOT_ON,1
```



```

WAIT: IF KEYIN(START_BUTTON) = 1 THEN GOTO WAIT

      SOUND PIEZO_BUZZER,80,8:DELAY 500
      SOUND PIEZO_BUZZER,80,8:DELAY 500
      SOUND PIEZO_BUZZER,60,15

      PWM L_MOTOR,SPEED_NORMAL
      PWM R_MOTOR,SPEED_NORMAL

LOOP: S0 = IN(SENSE_LEFT_IN)
      S1 = IN(SENSE_RIGHT_IN)
      IF S0 = 0 AND S1 = 1 THEN
          PWM L_MOTOR,SPEED_DOWN
          OUT LED_LEFT,1
      ELSEIF S0 = 1 AND S1 = 0 THEN
          PWM R_MOTOR,SPEED_DOWN
          OUT LED_RIGHT,1
      ELSE
          PWM L_MOTOR,SPEED_NORMAL
          PWM R_MOTOR,SPEED_NORMAL
          OUT LED_LEFT,0:OUT LED_RIGHT,0
      END IF
      GOTO LOOP

REM  [ FUNCTION ]-----

REM  [ END ]-----

```

Remarques:

A noter qu'il vous sera peut être nécessaire en fonction du niveau de courbure de votre parcours ou de l'état de la pile de modifier et d'adapter les vitesses de rotation des moteurs dans la déclaration des constantes (SPEED_NORMAL et SPEED_DOWN) afin que votre robot suive convenablement la ligne. De même dans certains cas extrêmes, il n'est pas exclu que les 2 moteurs du robot tournent à des vitesses légèrement différentes pour une même valeur de consigne (en raison des tolérances inhérentes à l'ensemble des composants rentant dans la composition du robot). Dans ce cas, il vous faudra instaurer des valeurs de consignes différentes pour les 2 moteurs pour la vitesse normale (SPEED_NORMAL) et la décélération de la roue (SPEED_DOWN) dans la déclaration des constantes et les lignes de programmes pilotant les moteurs (ceci faisant par ailleurs parti de l'initiation aux "joies" et aléas de la robotique !).

A noter qu'une fonction spéciale (**POKE &H12,&H34**) utilisée dans le listing permet de modifier un des pré diviseurs internes du PICBASIC-3B afin que le rapport cyclique des signaux PWM soit adapté au fonctionnement du robot. Cette instruction "joue" en fait directement sur les registres internes du microcontrôleur PICBASIC-3B (qui est un PIC16F876 ingénieusement programmé). A ce titre, il est impératif de ne pas modifier les valeurs de cette instruction qui peut (comme indiqué dans la notice des PICBASIC) être "fatale" pour le microcontrôleur (non pris en compte par la garantie) et qu'il faut donc ne pas utiliser en dehors des exemples donnés.

Dans tous les cas, si votre robot ne réagit pas correctement malgré votre programmation, commencez par vérifier l'état de la pile (utilisez de préférence une alimentation externe lors de vos mises au point).

ROBOT "VAGABOND Eddy"

Ce programme permet de concevoir un robot qui se déplacera tout seul sur une table sans tomber. Tant que les émissions infrarouges du robot sont réfléchies par la table et vues par ses capteurs, le robot recevra une information de niveau logique "haut" et il continuera à avancer. Lorsque l'avant du robot sera hors de la table (juste avant qu'il ne tombe), le niveau logique retourné sera "bas" (de part l'absence de réflexion) et suivant les cas de figures, le robot tournera à droite, à gauche ou reculera pour revenir sur la table. Bien évidemment, il faudra que la surface de la table soit "clair" afin que les émissions infrarouges puissent se réfléchir. Il est également possible de placer le robot sur une grande feuille blanche dans un périmètre délimité et entouré par une ligne noire (c'est d'ailleurs un essai préliminaire conseillé afin d'éviter que le robot ne tombe de la table suite à une vitesse excessive).

```
REM [DESCRIPTION DU PROGRAMME DISPO SUR LA DISQUETTE JOINTE]-----
' ROBOT "VAGADOND EDDY"
'A UTILISER SUR UNE SURFACE CLAIRE.

REM [ VARIABLE DECLARATION ]-----
DIM I AS BYTE
DIM J AS INTEGER
DIM S0 AS BYTE
DIM S1 AS BYTE
DIM S2 AS BYTE
DIM S3 AS BYTE
DIM MODE AS BYTE

REM [ CONSTANTS DECLARATION ]-----

CONST R_MOTOR = 9
CONST L_MOTOR = 10
CONST MOT_ON = 11
CONST PIEZO_BUZZER = 8
CONST START_BUTTON = 12

CONST STOP = 128
CONST FORWARD = 158
CONST BACKWARD = 78

CONST TURN_TIME = 250
CONST BACK_TIME = 200

CONST SENSE_LEFT_OUT = 4
CONST SENSE_LEFT_IN = 5
CONST SENSE_RIGHT_IN = 6
CONST SENSE_RIGHT_OUT = 7

CONST LED_LEFT = 13
CONST LED_RIGHT = 14
```



```

REM [ MAIN ]-----

POKE &H12,& H34 ' CHANGEMENT DU PREDIVISEUR DU SIGNAL PWM (1:16 > 1:1)

BEEP PIEZO_BUZZER
PWM R_MOTOR,STOP
PWM L_MOTOR,STOP
OUT MOT_ON,1

WAIT: IF KEYIN(START_BUTTON) = 1 THEN GOTO WAIT
SOUND PIEZO_BUZZER,80,8:DELAY 500
SOUND PIEZO_BUZZER,80,8:DELAY 500
SOUND PIEZO_BUZZER,60,15

LOOP: S0 = IN(SENSE_LEFT_OUT)
S1 = IN(SENSE_LEFT_IN)
S2 = IN(SENSE_RIGHT_IN)
S3 = IN(SENSE_RIGHT_OUT)
IF S0 = 0 AND S3 = 1 THEN
    OUT LED_LEFT,1:OUT LED_RIGHT,0
    GOSUB RIGHT_TURN
ELSEIF S0 = 1 AND S3 = 0 THEN
    OUT LED_LEFT,0:OUT LED_RIGHT,1
    GOSUB LEFT_TURN
ELSEIF S1 = 0 OR S2 = 0 THEN
    OUT LED_LEFT,1:OUT LED_RIGHT,1
    GOSUB U_TURN
ELSE
    OUT LED_LEFT,0:OUT LED_RIGHT,0
    PWM L_MOTOR,FORWARD
    PWM R_MOTOR,FORWARD
END IF
GOTO LOOP

REM [ FUNCTION ]-----

LEFT_TURN:
GOSUB BACK
PWM R_MOTOR,FORWARD
PWM L_MOTOR,BACKWARD
DELAY TURN_TIME
GOSUB MOTOR_STOP
RETURN

RIGHT_TURN:
GOSUB BACK
PWM R_MOTOR,BACKWARD
PWM L_MOTOR,FORWARD
DELAY TURN_TIME
GOSUB MOTOR_STOP
RETURN

U_TURN:
GOSUB BACK
PWM R_MOTOR,BACKWARD
PWM L_MOTOR,FORWARD
DELAY TURN_TIME
DELAY TURN_TIME
GOSUB MOTOR_STOP
RETURN

```

```

MOTOR_STOP:
    PWM R_MOTOR,STOP
    PWM L_MOTOR,STOP
    BEEP PIEZO_BUZZER
    RETURN

BACK: PWM R_MOTOR,BACKWARD
    PWM L_MOTOR,BACKWARD
    DELAY BACK_TIME
    GOSUB MOTOR_STOP
    RETURN

REM [ END ]-----

```

Remarques:

A noter qu'il vous sera intéressant de modifier et d'adapter les vitesses de rotation des moteurs dans la déclaration des constantes (FORWARD et BACKWARD et TURN_TIME et BACK_TIME) afin de modifier les réactions de votre robot (toutefois, prenez garde lors de vos essais à ce que le robot ne tombe pas de la table...).

A noter également qu'une fonction spéciale (**POKE &H12,&H34**) utilisée dans le listing permet de modifier un des pré diviseurs internes du PICBASIC-3B afin que le rapport cyclique des signaux PWM soit adapté au fonctionnement du robot. Cette instruction "joue" en fait directement sur les registres internes du microcontrôleur PICBASIC-3B (qui est un PIC16F876 ingénieusement programmé). A ce titre, il est impératif de ne pas modifier les valeurs de cette instruction qui peut (comme indiqué dans la notice des PICBASIC) être "fatale" pour le microcontrôleur (non pris en compte par la garantie) et qu'il faut donc ne pas utiliser en dehors des exemples donnés.

Dans tous les cas, si votre robot ne réagit pas correctement malgré votre programmation, commencez par vérifier l'état de la pile (utilisez de préférence une alimentation externe lors de vos mises au point).

TEST DES CAPTEURS LATERAUX

Les capteurs latéraux qui sont reliés à P0 - P3, fourniront une tension dont la valeur sera proportionnelle à la distance qui les séparent d'un obstacle. Le programme ci-dessous nécessite l'utilisation d'un afficheur LCD 2 x 16 caractères à commande série de type « ELCD162 », lequel devra être raccordé sur le connecteur spécifique du robot prévu à cet effet (il est préférable, sauf si vous utilisez une alimentation externe en remplacement de la pile de ne pas utiliser d'afficheur rétro éclairé en raison d'une consommation trop excessive). Lors de son exécution, le programme affichera différentes valeurs hexadécimales variant en temps réel en fonction de la distance entre un obstacle et chaque capteur.

```
' TEST DES CAPTEURS
' LES CAPTEURS SONT TESTES EN PARTANT DE LA GAUCHE.
' LES RESULTATS SONT MEMORISES APRES AVOIR CONVERSION DU FORMAT 10 A 8 BITS.

DIM SEN(4) AS BYTE
DIM TEMP AS INTEGER
DIM J AS BYTE

CONST TRIGGER = 15
CONST PIEZO_BUZZER = 8

BEEP PIEZO_BUZZER

SET PICBUS HIGH
LCDINIT
LOCATE 0,0
PRINT "SENSING DATA"
10  FOR J = 0 TO 3
      OUT TRIGGER,1           ' L'EMISSION IR DURE 200us (APPROXIMATIVEMENT)
      SEN(J) = ADIN(J) >> 2
      OUT TRIGGER,0
      DELAY 20
    NEXT J
    LOCATE 0,1
    FOR J = 0 TO 3
      PRINT HEX(SEN(J),2);" "
    NEXT J

    GOTO 10
```

Le robot BASIC-BOT offre de nombreuses possibilités de réalisation que nous vous laissons le soins de réaliser (c'est après tout le but et le rôle de ce robot). Vous pourrez ainsi aisément le programmer pour réaliser :

- Un robot capable de se déplacer tout seul en évitant les obstacles qui se dressent devant lui grâce à ses capteurs latéraux.
- Un robot capable de se déplacer seul dans un labyrinthe.
- 2 robots capables de se suivre (le premier évoluant en arrière, et le second susceptible de le suivre à la trace grâce à une communication infrarouge à établir entre les 2 BASIC-BOT, faisant office de ligne conductrice.
- Une base infrarouge susceptible à un robot de revenir à cette dernière tout seul, etc, etc...

Si vous avez des difficultés à réaliser de telles applications, vous trouverez sur la disquette jointe un 4^{ème} programme «avorder» (non décrit dans cette notice mais très simple à comprendre). Ce dernier a été réalisé sur la base du programme du robot «vagabond» mais avec une détection issue de 2 des 4 capteurs IR latéraux afin de réaliser un robot mobile capable de détecter les obstacles qui se dressent devant lui et de les contourner. Vous pourrez bien évidemment le compléter et l'améliorer en testant l'état des 4 capteurs latéraux afin de faire réagir le robot en conséquence.

Si vous avez réalisé des programmes d'applications intéressants pour le BASIC-BOT, n'hésitez pas à nous les faire parvenir afin que nous puissions les mettre à la disposition (avec le nom de son auteur) des autres utilisateurs via la notre site Internet.

Schéma théorique complet (1/2)

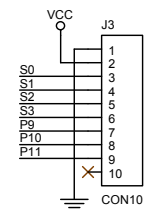
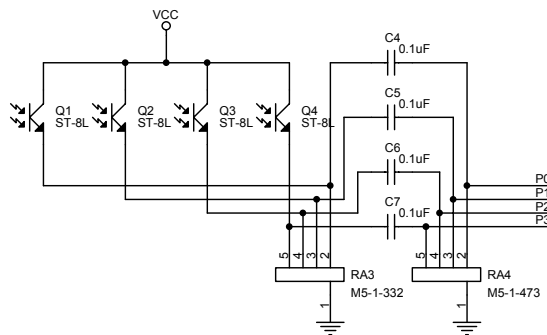
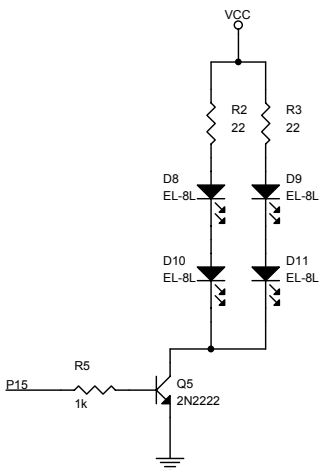
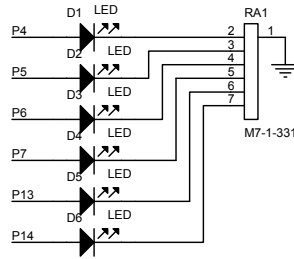
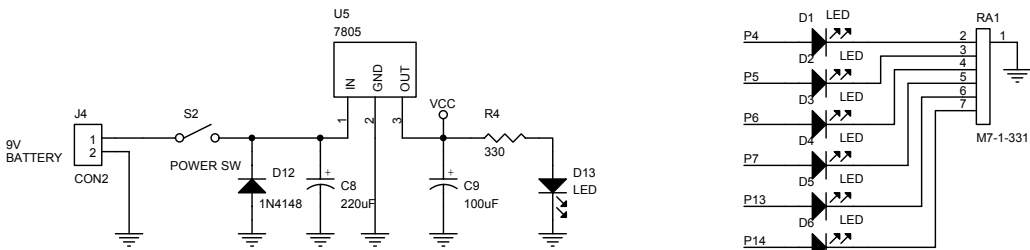
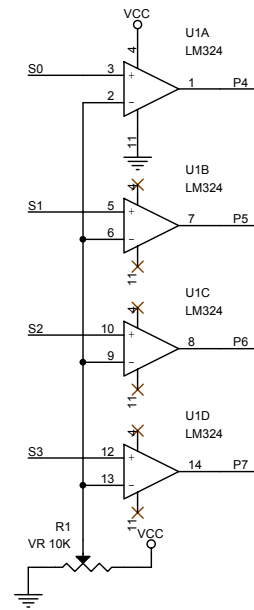
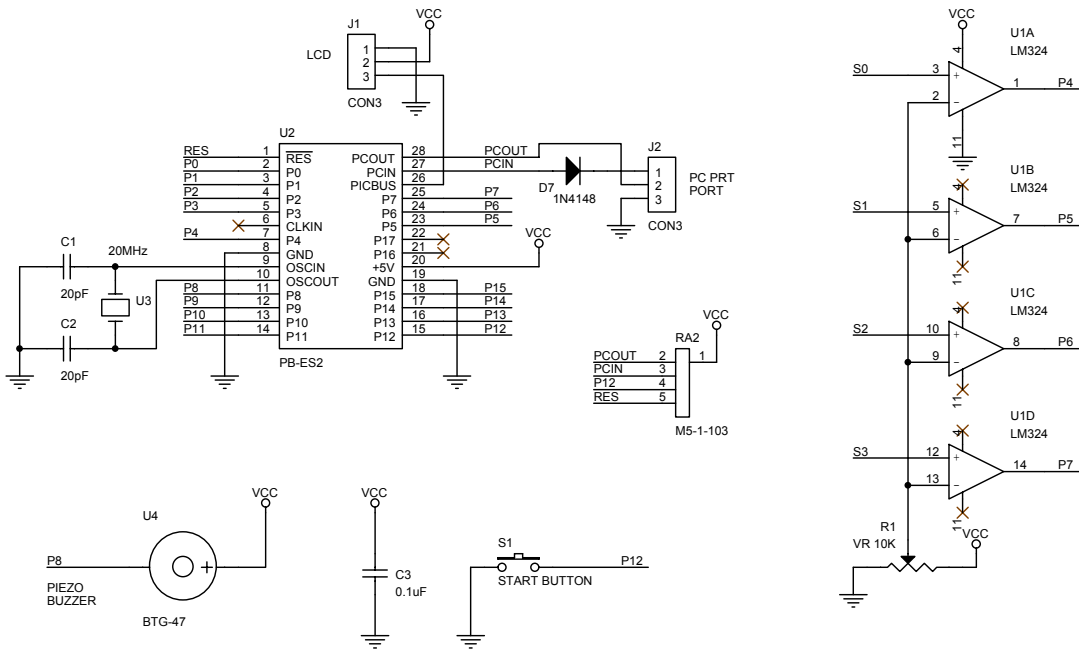


Schéma théorique complet (2/2)

