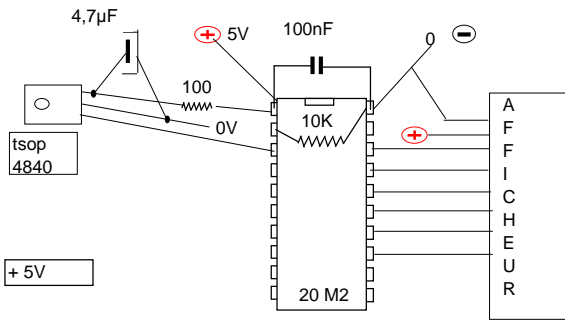
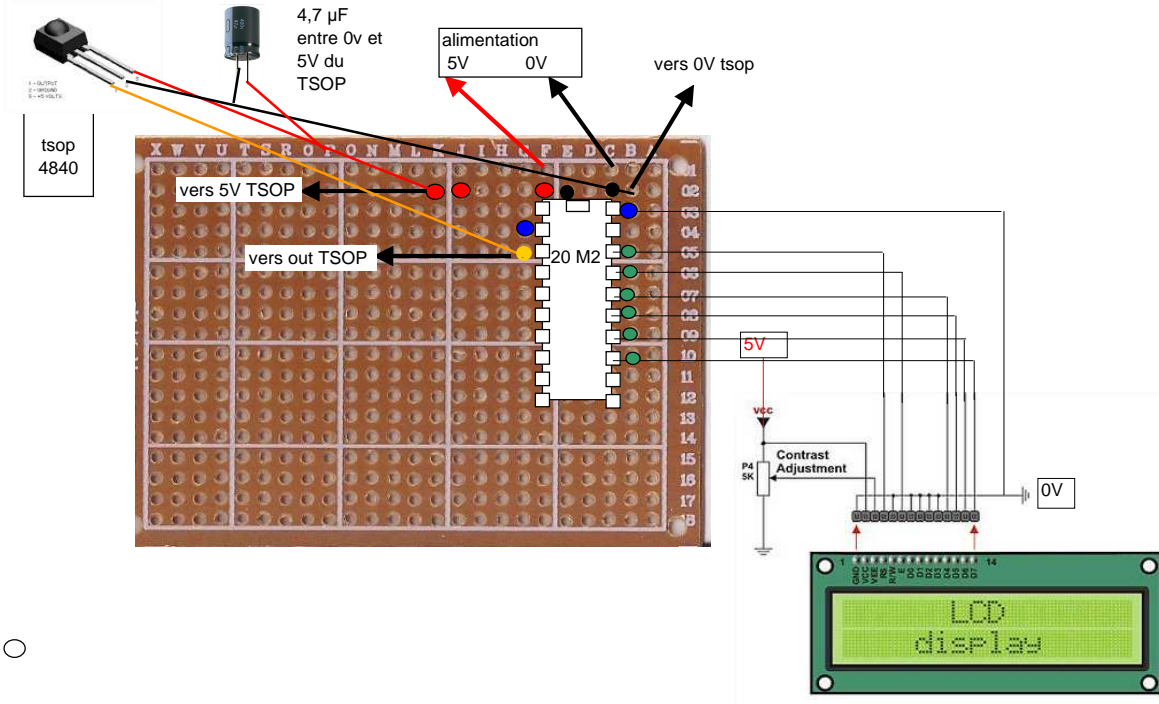


Affichage du code IR sur LCD



composants	connection
20M2	F3 patte 1 F4 patte 20 ... C3 patte 20 C4 patte 19....
● condensateur 100nF	F2 et C2
● resistance 10K	G4 et B3
● résistance 100 ohms	entre F2 et J2
TSOP	
● patte out du TSOP	relié à G5
0V tsop	relié à B2
5V tsop	relié à F2
● CONNECTION LCD de B0 à B5 Voir ci dessus	

```

#picaxe 20M2 'directive 20M2
setfreq m4 '4MHz pour respect des temporisations
dirsB=%11111111 'défini les ports B en sorties
dirsc=%00000000 'défini les ports B en entrées
SYMBOL RS = B.0 ; 0 = Command 1 = Data
SYMBOL E = B.1 ; 0 = Idle 1 = Active
SYMBOL DB4 = B.2 ; LCD Data Line 4
SYMBOL DB5 = B.3 ; LCD Data Line 5
SYMBOL DB6 = B.4 ; LCD Data Line 6
SYMBOL DB7 = B.5 ; LCD Data Line 7
SYMBOL RSCMDmask = %00000000 ; Select Command register
SYMBOL RSDATmask = %00000001 ; Select Data register
SYMBOL cpt = b11
SYMBOL char = b12
SYMBOL rsbit = b13
SYMBOL code = b14
DATA 0,( $33 ) ; %0011---- %0011---- 8-bit / 8-bit
DATA 1,( $32 ) ; %0011---- %0010---- 8-bit / 4-bit
DATA 2,( %00101000 ) ; %001LNF00 Display Format
DATA 3,( %00001100 ) ; %00001DCB Display On
DATA 4,( %00000110 ) ; %0000011S Cursor Move
DATA 5,( $01 ) ; Clear Screen
DATA 6,("code=")
code=0
PowerOnReset:
FOR cpt = 0 TO 5
READ cpt,char
GOSUB SendInitCmdByte
NEXT
symbol dcode = b1
symbol ucode = b2
symbol ccode= b3
char= $80 | $00 ' curseur début ligne1
gosub SendCmdByte
FOR cpt = 6 TO 10 'lecture texte "code="
READ cpt,char
GOSUB SendDataByte
NEXT
gosub trans
gosub ecrit
DisplayTopLine:
debut:
irin [100,debut],c.7,code
char= $80 | $00 ' curseur début ligne1
gosub SendCmdByte
FOR cpt = 6 TO 10 'lecture texte "code="
READ cpt,char
GOSUB SendDataByte
NEXT

```

```

gosub trans
gosub ecrit
pause 20
goto DisplayTopLine
trans:
bintoascii code,ccode,dcode,ucode
return
ecrit: 'on écrit de gauche à droite
char=ccode
GOSUB SendDataByte
char=dcode
GOSUB SendDataByte
char=ucode
GOSUB SendDataByte
return
SendInitCmdByte:
PAUSE 15 ; Delay 15mS
SendCmdByte:
rsbit = RSCMDmask ; Send to Command register
SendDataByte:
pinsb = char & $F0 / 4 | rsbit ; Put MSB out first
PULSOUT E,1 ; Give a 10uS pulse on E
pinsb = char & $0F * 4 | rsbit ; Put LSB out second
PULSOUT E,1 ; Give a 10uS pulse on E
rsbit = RSDATmask ; Send to Data register next
RETURN

```