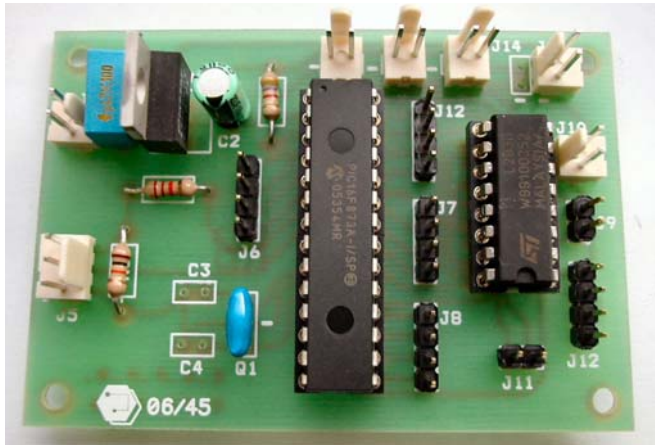


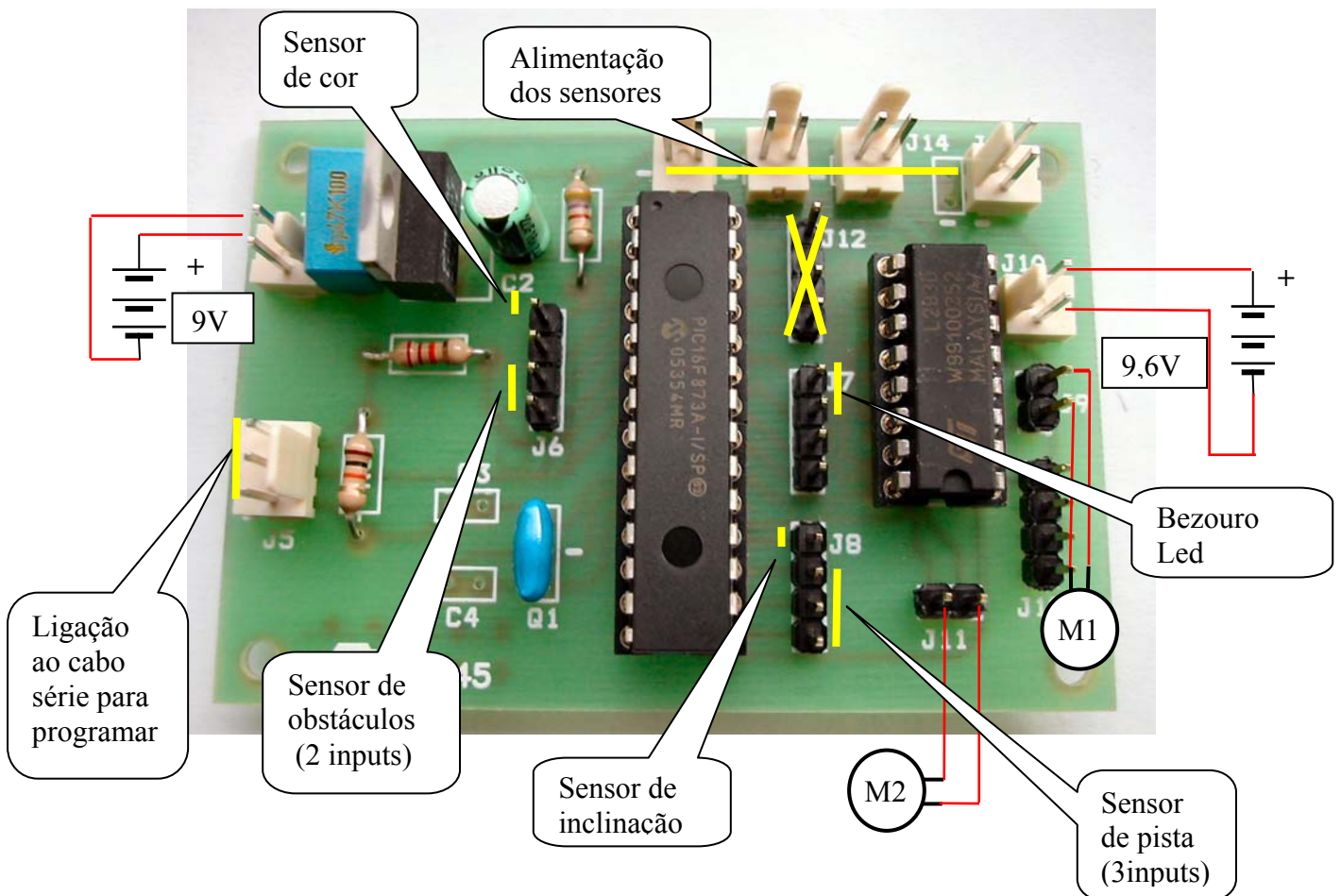
DEAR Robot 2006/7

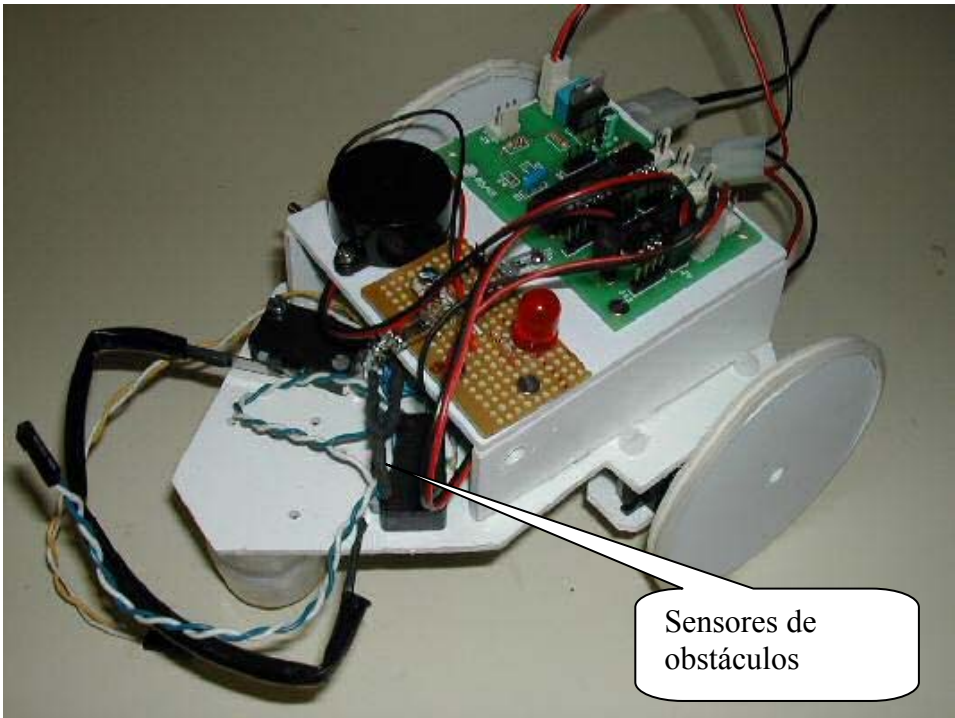
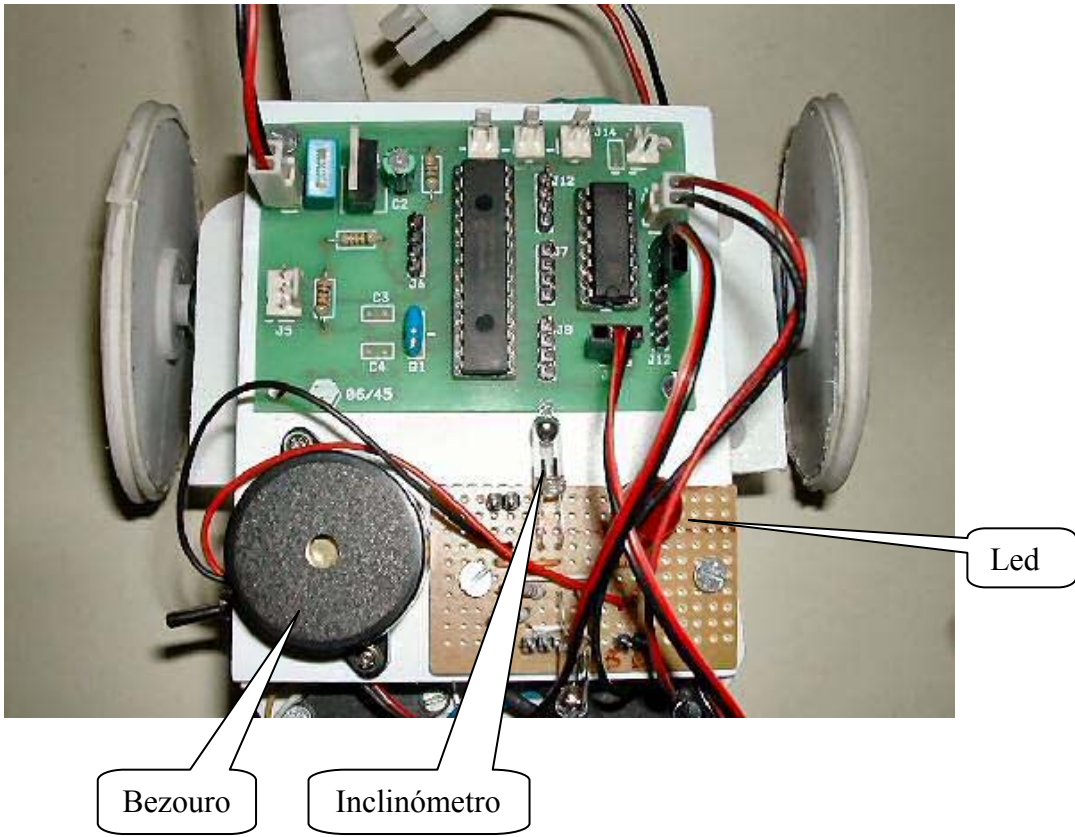


DEAR ROBOT 2006/7

Ligações

Sugere-se o seguinte esquema de ligações de sensores:





Testes Iniciais de Programação

1. Ligações

- Motor esquerdo (M1)
output5 (pino 26)
output4 (pino 25)
pwm2 (pino13)
- Motor direito (M2)
output7 (pino 28)
output6 (pino 27)
pwm2 (pino 12)
- Sensores de pista
input6– Esquerdo (pino 17)
input5– Central (pino 16)
input4– Direito (pino 15)
- Sensor de obstáculos (bumpers)
input portA 2 – microinterruptor esquerdo (pino 4)
input portA 3 – microinterruptor direito (pino 5)
- Avisador Bezouro
output3 (pino 24)
- Avisador LED
output2 (pino 23)
- Sensor de Cores (LDR)
ADC0 (pino 2)
- Sensor de inclinação
input7 (pino 18)

2. Procedimentos para começar a programar:

- Ligar o computador e executar o software Picaxe Programming Editor
 - Opções de PICAXE28X e 4MHz
 - Opção de porto série COMx
- Ligar o cabo série entre o PICAXE e o PC.
- Escrever o primeiro programa.
- Ligar a alimentação da placa PICAXE.
- Transferir o programa.

3. Primeiro programa

- Testar os motores e registar o sentido de rotação

Para isso vamos começar pelo motor esquerdo:

```
‘*****
```

```
início:  
  
    high 5  
    low 4  
    pwmout 2, 10, 40  
  
goto início
```

```
‘*****
```

- Transferir e verificar!! Roda para a frente ou para trás? Registrar.
- Alterar o programa para

```
‘*****
```

```
início:  
  
    low 5  
    high 4  
    pwmout 2, 10, 40  
  
goto início
```

- Deverá inverter a rotação do motor. Registrar.
- Fazer o mesmo para o motor direito.

```
‘*****
```

```
início:  
  
    high 7  
    low 6  
    pwmout 1, 10, 40  
  
goto início
```

```
‘*****
```

- Registrar! Registrar! Registrar!

4. Segundo programa – rodopiar

```
*****
inicio:
    gosub direita
    pause 1000
    gosub esquerda
    pause 1000
    gosub avancar
    pause 1000
    gosub recuar
    pause 1000
goto inicio
*****
avancar:
    high 7
    low 6
    high 5
    low 4
    pwmout 1,10,30
    pwmout 2,10,30
    return
*****
direita:
    low 7
    low 6
    high 5
    low 4
    pwmout 1,10,30
    pwmout 2,10,30
    return
*****
esquerda:
    high 7
    low 6
    low 5
    low 4
    pwmout 1,10,30
    pwmout 2,10,30
    return
*****
recuar:
    low 7
    high 6
    low 5
    high 4
    pwmout 1,10,30
    pwmout 2,10,30
    return
```

5. Terceiro programa - sons

‘*****

```
symbol bezouro = 3
inicio:

    sound bezouro,(50,50)
    sound bezouro,(100,50)
    sound bezouro,(120,50)
    pause 300
goto inicio
```

‘*****

6. Quarto programa - sensor de obstáculos

```
symbol Esq = input2
symbol Dir = input3
symbol vel = b0

    let vel=28
inicio:

    if porta Esq = 0 and Dir = 0 then frente
    if porta Esq = 0 and Dir = 1 then desvio_esquerda
    if porta Esq = 1 and Dir = 0 then desvio_direita
    if porta Esq = 1 and Dir = 1 then desvio_frente
    goto inicio

frente:
    gosub avancar
    goto inicio

desvio_esquerda:
    gosub recuar
    pause 400
    gosub esquerda
    pause 200
    goto inicio

desvio_direita:
    gosub recuar
    pause 400
    gosub direita
    pause 200
    goto inicio
```

```
desvio_frente:
```

```
    gosub recuar  
    pause 500  
    gosub direita  
    pause 300  
    goto inicio
```

```
*****
```

```
avancar:
```

```
    high 7  
    low 6  
    high 5  
    low 4  
    pwmout 1,10,vel  
    pwmout 2,10,vel  
    return
```

```
*****
```

```
direita:
```

```
    low 7  
    low 6  
    high 5  
    low 4  
    pwmout 1,10,vel  
    pwmout 2,10,vel  
    return
```

```
*****
```

```
esquerda:
```

```
    high 7  
    low 6  
    low 5  
    low 4  
    pwmout 1,10,vel  
    pwmout 2,10,vel  
    return
```

```
*****
```

```
recuar:
```

```
    low 7  
    high 6  
    low 5  
    high 4  
    pwmout 1,10,vel  
    pwmout 2,10,vel  
    return
```

Note que na instrução *if* tivemos que introduzir a expressão **porta** para indicar que estamos a usar os pinos 4 e 5 como inputs digitais. Os pinos 2 a 5 são pinos analógicos ou digitais consoante a programação.

7. Quinto programa - sensor de pista

Em primeiro lugar temos que testar os sensores e confirmar os pinos de entrada.

Para isso, pode usar-se a instrução *debug*:

```
*****
```

```
início:
    b0 = pins
    debug
    goto início
```

A janela de variáveis mostra em binário o estado das entradas.

Basta ir colocando uma fita preta em frente de cada um dos sensores e registrar qual o pino.

%76543210

%01000000 significa que a entrada 6 está alta (preto)

%01100000 significa que as entradas 6 e 5 estão altas (preto)

Registrar!!!!!!

Programa para seguimento de pista com três sensores.

```
*****
```

```
'Quinto programa
' 0 - branco
' 1 - preto

symbol dir = pin6
symbol ct = pin5
symbol esq = pin4
symbol vel = b0

    vel = 35
início:

    if esq = 0 and ct = 0 and dir = 0 then início                   'nota 1
    if esq = 0 and ct = 0 and dir = 1 then virar_direita
    if esq = 0 and ct = 1 and dir = 0 then frente
    if esq = 0 and ct = 1 and dir = 1 then virar_direita_rapido
    if esq = 1 and ct = 0 and dir = 0 then virar_esquerda
    if esq = 1 and ct = 0 and dir = 1 then frente                   'nota 2
    if esq = 1 and ct = 1 and dir = 0 then virar_esquerda_rapido
    if esq = 1 and ct = 1 and dir = 1 then parar                   'nota 3
    goto início
```

```
frente:
    gosub avancer
    goto inicio
virar_direita:
    gosub direita
    goto inicio
virar_esquerda:
    gosub esquerda
    goto inicio

virar_direita_rapido:
    gosub direita_r
    goto inicio

virar_esquerda_rapido:
    gosub esquerda_r
    goto inicio
```

```
avancar:
    high 7
    low 6
    high 5
    low 4
    pwmout 1,10,vel
    pwmout 2,10,vel
    return
```

```
direita:
    low 7
    low 6
    high 5
    low 4
    pwmout 1,10,vel
    pwmout 2,10,vel
    return
```

```
esquerda:
    high 7
    low 6
    low 5
    low 4
    pwmout 1,10,vel
    pwmout 2,10,vel
    return
```

```
direita_r:
    low 7
```

```

    high 6
    high 5
    low 4
    pwmout 1,10,vel
    pwmout 2,10,vel
    return
*****
esquerda_r:
    high 7
    low 6
    low 5
    high 4
    pwmout 1,10,vel
    pwmout 2,10,vel
    return

*****
parar:
    low 7
    low 6
    low 5
    low 4
    pwmout 1,10,vel
    pwmout 2,10,vel
    return

```

Notas:

1. A situação **esq = 0 and ct = 0 and dir = 0** corresponde a estarem os três sensores no branco. Esta situação é muito frequente pois a distância entre sensores é fixa e quase igual à largura da pista. Só há uma hipótese viável que é fazer avançar.
2. A situação **esq = 1 and ct = 0 and dir = 1** corresponde a ter os sensores esquerdo e direito em cima da linha e o central no branco. Não parece possível isso acontecer pelo que se programa para avançar.
3. A situação **esq = 1 and ct = 1 and dir = 1** corresponde a ter os três sensores em cima da pista preta. Esta situação só acontece no T da chegada. Pode acontecer na subida da rampa por afastamento excessivo dos sensores em relação à pista.

8. Sexto programa - Sensor analógico LDR

O LDR vai estar num divisor potenciométrico em série com uma resistência de 1k2 e ligado entre +5V e a massa. Isto é, a tensão que entra no pino ADC0 (ponto comum entre o LDR e a resistência) é uma tensão que pode variar consoante a luz incidente aproximadamente entre 4V e 1V.

Se usarmos a instrução *readadc10*, o intervalo vai variar entre 0 (para 0V) e 1024 (para 5V).

Programa para ler o LDR e apitar quando há falta de luz.

```
‘*****
```

```
'Sexto programa
'programa LDR

symbol leitura = w0
symbol led = 2
symbol bezouro = 3

inicio:

    readadc10 0,leitura

    if leitura < 650 then alarme
    goto inicio

alarme:
    sound bezouro,(50,50)
    high led
    sound bezouro,(100,50)
    low led
    sound bezouro,(120,50)
    pause 300
    goto inicio
```

```
‘*****
```

9. Sétimo programa - Sensor inclinação

O sensor de inclinação está ligado em série com uma resistência de 4k7 e ligado entre +5V e a massa. O ponto médio entre a resistência e o sensor está ligado ao input7.

Programa para ler o inclinómetro e apitar quando detecta a rampa.

```
‘*****
```

```
‘programa Inclinometro

symbol rampa = pin7
symbol led = 2
symbol bezouro = 3

inicio:

    if rampa=1 then alarme
    goto inicio
```

```
alarme:  
  sound bezouro,(50,50)  
  high led  
  sound bezouro,(100,50)  
  low led  
  sound bezouro,(120,50)  
  pause 300  
  goto inicio
```

Bom!! Se tudo deu certo até agora, é tempo de começar a pensar no programa a sério!!
Bom trabalho!!!