

```

/* *****
 * SERVO ET LCD *
*****

Un potentiomètre donne l'angle du servo-moteur
La LCD affiche l'angle

LE CIRCUIT

LCD:
LCD RS pin to digital pin 12
LCD Enable pin to digital pin 11
LCD D4 pin to digital pin 5
LCD D5 pin to digital pin 4
LCD D6 pin to digital pin 3
LCD D7 pin to digital pin 2
LCD R/W pin to ground
10K resistor:
ends to +5V and ground
wiper to LCD VO pin (pin 3)

Potentiomètre:
Entre le +5 et la masse.
Curseur sur l'entrée analogique A0 (pin 14)

Poussoir:
Pin IN 10

*/

// Inlu les librairie:
#include <LiquidCrystal.h>
#include <Servo.h>
#include <EEPROM.h>

int SensorPin = A0;
int SensorValue = 0;
int Azimuth=0;
int ads = 0;
Servo myservo;// create servo object to control a servo
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

```

```

// définit le nombre de colonnes et de lignes du LCD:
lcd.begin(16, 1);
// Ecrit sur LCD:
lcd.print("Azimuth:");

myservo.attach(9); // attaches the servo on pin 9 to the servo object

pinMode(10,INPUT);
}

void loop() {

//Lit dernier azimuth en mémoire
Azimuth=EEPROM.read(0);

//Ecrit la valeur de l'angle sur la LCD
lcd.setCursor(9,0);
lcd.print(Azimuth);

//Ecrit la valeur de l'angle sur le servo
myservo.write(Azimuth);
delay(50);

do // tourne tant que bouton pas pressé
{

} while (digitalRead(10)); // bouton entrée dig 10 pas pressé

//Efface la valeur précédente
lcd.setCursor(9, 0);
lcd.print("    ");

//Lit la valeur du potentiomètre (entre 0 et 1023)
SensorValue =analogRead (SensorPin);

//Convertit la valeur lue en angle (entre 0 et 179 degres)
Azimuth=map(SensorValue,0,1023,0,179);

//Ecrit la valeur de l'angle sur la LCD
lcd.setCursor(9,0);
lcd.print(Azimuth);

//Mémorise l'angle
EEPROM.write(ads,Azimuth);

```

```
//Erit la valeur de l'angle sur le servo
myservo.write(Azimuth);
delay(50);

/* // HI en morse
tone(6,800,100);
  delay(200);
tone(6,800,100);
delay(200);
tone(6,800,100);
delay(200);
tone(6,800,100);
delay(300);
tone(6,800,100);
  delay(200);
tone(6,800,100);
  delay(200);
*/
// AZ en morse
tone(6,800,100); //passe à l'instruction suivante mais le ton continue
delay(200);      // = 100ms ton et 100ms silence
tone(6,800,300);
delay(500);
tone(6,800,300);
delay(400);
tone(6,800,300);
delay(400);
tone(6,800,100);
delay(200);
tone(6,800,100);
delay(200);

}
```