Ex1

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int tmp = 5;

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.print("Hello Arduino");

lcd.setCursor(1, 1);

lcd.print(tmp);

}

void loop() {

}

Ch1

// include the library code:

#include <LiquidCrystal.h>

// these constants won't change. But you can change the size of

// your LCD using them:

const int numRows = 2;

const int numCols = 16;

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(numCols,numRows);

}

void loop() {

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Tanakorn Masran");

// loop from ASCII 'a' to ASCII 'z':

for (int thisCol = 0; thisCol <= 9; thisCol++) {

// set the cursor position:

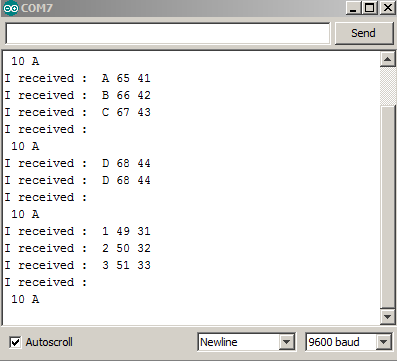
lcd.setCursor(thisCol,1);

lcd.print(thisCol);

delay(200);

}

}



/\* Read a text received on serial arduino port and display on LCD\*/

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define INLENGTH 16

#define INTERMINATOR 13

char inString[INLENGTH+1];

int inCount;

int incomingByte = 0;

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

// prints title with ending line break

Serial.println("Reads incoming serial data and display on LCD.");

//pinMode(13, OUTPUT);

//digitalWrite(13, HIGH); // set the LED on

lcd.setCursor(0,0); //col, row

lcd.println("Ready to receive");

}

void loop()

{

inCount = 0;

do

{

while (!Serial.available()); // wait for input

inString[inCount] = Serial.read(); // get it

Serial.print(inString[inCount]);

if (inString[inCount] == INTERMINATOR) {

inString[inCount] = 0;

break;

}

++inCount;

} while(inCount < INLENGTH);

lcd.clear();

lcd.setCursor(0,0); //col, row

lcd.println("Ready to receive");

lcd.setCursor(0,1); //col, row

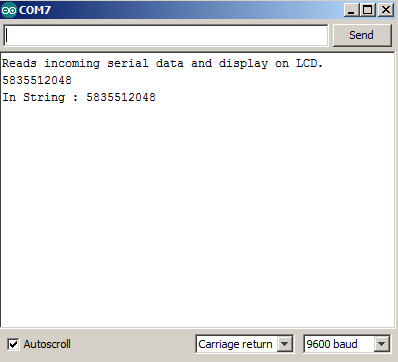
lcd.print(inString);

Serial.print("\nIn String : ");

Serial.println(inString);

inString[0] = 0; // null terminate the string

}



Chk3

/\* Read a text received on serial arduino port and display on LCD\*/

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int ss=00;

int mm=00;

int hh=00;

int seconds;

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

// prints title with ending line break

lcd.setCursor(0,0); //col, row

lcd.print("HH:MM:SS");

}

void loop()

{

seconds = millis()/100;

lcd.setCursor(0,1); //col, row

hh = seconds/3600%24;

if(hh<10)

lcd.print(0);

lcd.print(hh+(seconds/3600%24));

lcd.setCursor(2,1); //col, row

lcd.print(':');

mm = seconds/60%60;

lcd.setCursor(3,1); //col, row

if(mm<10)

lcd.print(0);

lcd.print(mm);

lcd.setCursor(5,1); //col, row

lcd.print(':');

ss = seconds%60;

lcd.setCursor(6,1); //col, row

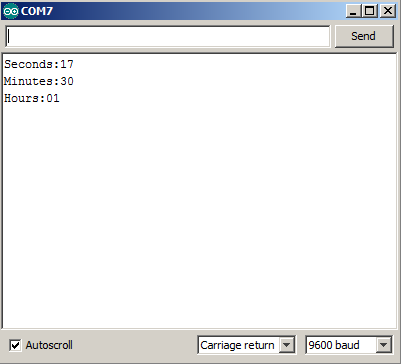
if(ss<10)

lcd.print(0);

lcd.print(ss);

}

Chk4



/\* Read a text received on serial arduino port and display on LCD\*/

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int ss=0;

int mm=0;

int hh=0;

long seconds;

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

// prints title with ending line break

lcd.setCursor(0,0); //col, row

lcd.print("HH:MM:SS");

}

void loop()

{

if(seconds==(60\*60\*24))

seconds = 0;

seconds = millis()\*10;

lcd.setCursor(0,1); //col, row

hh = seconds/3600%24;

if(hh<10)

lcd.print(0);

lcd.print(hh);

lcd.setCursor(2,1); //col, row

lcd.print(':');

mm = seconds/60%60;

lcd.setCursor(3,1); //col, row

if(mm<10)

lcd.print(0);

lcd.print(mm);

lcd.setCursor(5,1); //col, row

lcd.print(':');

ss = seconds%60;

lcd.setCursor(6,1); //col, row

if(ss<10)

lcd.print(0);

lcd.print(ss);

if(Serial.available())

{

char inChar = Serial.read();

if(inChar == 'h')

{

Serial.print("Hours:");

if(hh<10)

Serial.print(0);

Serial.println(hh);

}

else if(inChar == 'm')

{

Serial.print("Minutes:");

if(mm<10)

Serial.print(0);

Serial.println(mm);

}

else if(inChar == 's')

{

Serial.print("Seconds:");

if(ss<10)

Serial.print(0);

Serial.println(ss);

}

}

}