

Digital Advertising Tag System

A thesis submitted to the
Faculty of the Computer Engineering Technology Program
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of

Bachelor of Science

in Computer Engineering Technology
at the College of Engineering & Applied Science

by

RICHARD TOLSON

Bachelor of Science University of Cincinnati

May 2011

Faculty Adviser: Michael Haas

25 S. Lake rd.
Glendale, Ohio 45246

June 7, 2011

ECET Faculty
University of Cincinnati
College of Applied Science
2220 Victory Parkway
Cincinnati, Ohio 45206

Dear ECET Faculty

Attached is the final report for my senior design project “Digital Advertising Tag System” which was requested by Professor Michael Haas with a deadline of June 7th 2011.

This report outlines the current problems with how convenience stores display and update pricing information every week. This report also proposes the Digital Advertising Tag System as the solution to this problem. This report explains the Digital Ad-Tag System concept and technical specifications as well as showing the circuit diagram and flow chart created during the design process. The focus of this project was to create an array of Graphical LCD screens which would be used as a shelf edge pricing system for convenience stores.

Thank you for your time and consideration. If you have any questions please feel free to call me at (513) 827-0774 or contact me through email at rjtolson@yahoo.com.

Sincerely,

Richard Tolson

DEDICATION

I would like to dedicate this Senior Design Report to the faculty and staff of the University of Cincinnati College of Applied Science Department of Electrical and Computer Engineering for being helpful and supportive of the students in this program for the passed five years, even through uncertain and confusing times.

Thank you all for of your time and effort.

TABLE OF CONTENTS

Abstract.....1

Discussion.....2

 Problem.....2

 Solution.....2

 Timeline.....5

 Budget.....6

 Problems Encountered.....7

 Future Enhancements.....8

Conclusion.....9

References.....10

Appendix A.....11

Appendix B.....19

ABSTRACT

Everyone has been to a convenience store at one time or another and have seen that these stores are arranged into groups of Isles containing shelves where the products are displayed. Adhesive labels are placed at the edge of the shelf in order to inform customers of the current price of the product just above it on the shelf. Traditionally these labels would be removed and replaced once a week when the product's price has changed or a sale is running on that item. Over time the cost for the amount of new weekly advertising tags becomes very large. The Digital Advertising Tag System solves this problem by replacing the product pricing labels at the edge of the convenience store's shelf with a Graphical LCD Module which will display product prices, Sale information, and Bitmap images. The Displays will be driven by an Arduino development kit containing an Atmega328P microcontroller which will allow the displays to be remotely and automatically updated with little possibility for human error. The Digital Ad-Tag System will eliminate the need to print out weekly price change and Advertising labels which will save the store much time and money.

DISCUSSION

PROBLEM

Between 2003 and 2006 I was working in the photo lab of the convenience store chain Walgreen's. While I was there one of my weekly tasks would be to "do the Ad". This would involve walking up and down the store aisles assigned to me and remove the old advertising stickers from the previous week. Then I would find the stack of about twenty pages of new advertising labels which belonged to those same aisles and proceed to tear them apart across perforated lines because many labels were printed onto one sheet. Finally I would walk back through the aisles and place the new labels at the shelf edge for the appropriate products. After a few weeks of this I started to realize how many mylar and paper advertising labels our store actually goes through, then I thought about all the stores across the country doing the same thing and I knew there had to be a better way. Another task that I would have been assigned to do from time to time was to do a "Product Reset". This is where all the products on a column of shelves would be rearranged with some products being removed from the facing and some new products being added. In order to accomplish this we would have to print off the "Plan-o-gram" information which was usually just a few pages from a PDF document which held the visual layout of each shelf and the products they need to contain. These Plan-o-grams showed filled in blocks representing the dimensions of the products that are to be placed on the shelf as well as the Inventory code for each product which tells the employee where in the stock room the product is held. Along with the plan-o-grams there are about 4 pages (more with larger resets) of new pricing labels which need to be attached to the edge of the shelf with their corresponding products. After a while I started thinking about this situation as well and realized that there may be an answer to this problem.

SOLUTION

After thinking about the problem for a while I decided that the solution here would be to create a display that would run the whole length of the edge of a convenience store shelf which should be able to display prices and product information as well as weekly sale information in full color. A push button would provide the functionality of allowing the employee to switch a column of shelves or even the entire aisle over to a "Plan-O-Gram" mode which will display graphically how to reset the shelf, this would tell the employee how wide each new product will be and where they will need to be placed on the shelf as well as where the products are stored in the stockroom. If developed the Digital Ad-Tag System could revolutionize the process currently in place of updating and adjusting product pricing for convenience stores around the world. This system will also cut down on the need for paper pricing labels, Sale tags, and printing Plan-o-gram sets, as well as all the ink involved.

To Demonstrate this device I created a proof of concept design for the Digital Ad-Tag System. The design consists of four GDM12864HLCM Liquid Crystal Display Modules measuring 75mm (l) x 52.7mm (w) x 10mm (h) lined up side by side to create a display that is 1 foot in length. The GLCD modules were purchased from Sparkfun for \$19.95 each and have a viewing area of 64x128 pixels. Each GLCD Display has twenty interface pin connections which are necessary for the display to operate properly. On each display pin 1 was wired to a 5 volt source, pin 2 was wired to Ground, pin 3 connects to the wiper arm of a 10K ohm Potentiometer (one for each screen) this is used for contrast adjustment on the GLCD module, pins 4 – 11 correspond to DB0 – DB7 and are the parallel data inputs to each display. Each Display contains two KS0108

64CH segment drivers used for dot matrix LCD's, pin 12 is the CS2 pin which activates the second KS0108 segment driver, pin 13 is the CS1 activates the first KS0108 segment driver. Pin 14 is /RES which will reset the displays after they have been updated, pin 15 is the R/W where setting this pin to a logic high or logic low will tell the drivers to read from the display or write to the display. Pin 16 is the D/I control where a logic high will tell the driver to accept data and a logic low will tell the driver to accept instructions. Pin 17 is the enable pin, if this pin is set low the GLCD module will not display anything but the modules memory still has all the loaded data so when the enable pin would be set high the displayed information will be shown again. Pin 18 is the the negative bias to the potentiometer used for the GLCD Contrast, this must be attached to the same potentiometer used for pin 3 of the same display. Pin 19 is designated as A and is responsible for supplying a little less than 5 volts for the backlight, to do this I placed a 36 ohm resistor between the 5 volt source and pin 19. Pin 20 is the ground for the GLCD backlight and can be tied to the same connection as pin 2. Figure 1 shows the completed GLCD array.



Figure 1: Digital Ad-Tag Display Modules

The array of GLCD modules was then connected to a microcontroller which would be responsible for updating the graphical information on each display. There were some issues with choosing the proper microcontroller for this application, the details are explained in the problems encountered section of this document. I finally decided to go with the Arduino UNO development kit which included the interface board with the ATmega328P microcontroller, a small breadboard, a dozen or so pre-tinned wire, an a handbook of examples for using the development kit. The Atmel ATmega328P Microcontroller allows for twenty-three programmable I/O lines and 32 Kilobytes of memory which is a fairly large amount of space for this application. The ATmega328P is placed into the Arduino UNO board which provides easily accessible pin connections to fourteen digital input/output pins and six analog inputs, and includes a 16 MHZ crystal oscillator, a USB connection, a power jack, and everything needed to support the microcontroller. The Arduino UNO also supplies the 5 volts necessary to power the array of GLCD modules. The Arduino UNO board is 68.58mm (l) x 53.34mm (w). the 5 volt output from the Arduino was connected to the power rail of a small breadboard where it would provide the necessary 5 volt input for each GLCD module as well as the supply voltage for the four 10K ohm potentiometers, four 36 ohm resistors, and power for a 7408 Quadruple 2-input AND gate which is used to multiplex the array of GLCD modules. Arduino pins 8,9,10, and 11 are connected to pins D0, D1, D2, and D3 and Arduino pins 4,5,6,and 7 are connected to pins D4,D5, D6, and D7 of all four of the GLCD Modules. Analog pins A0, A1, A2, and A3, connect to to the CS1, CS2, /RES, R/W, and D/I pins respectively for all four of the GLCD modules. Pin

A5 will act as an input from a push button switch, this switch will trigger the microcontroller to switch the images displayed on the GLCD modules between showing product pricing information and displaying the employee use “Plan-O-Gram” mode. Analog pin A4 provides the enable signal and is connected to one input of each of the four AND gates of the 7408 Chip. The remaining inputs of the AND gates are connected to Arduino pins 12, 13, 3, and 2. the coding within the microcontroller will assign these pins to to GLCD module 1, 2, 3, and 4 respectively, the outputs of the corresponding AND gates will connect to the enable pin (pin 17) of the appropriate GLCD module. If a logical high is sent out any of these pins it will be AND'ed with the logical high value of the Enable pin (A4) and will successfully activate the display. This is used to multiplex the data being sent to the GLCD modules so only one display is actively listening to the data and control lines while the other three are disabled. In this way unique information can be sent to each GLCD independently. Figure 2 shows the surface of the Arduino UNO board with all previously mentioned port pins labeled.

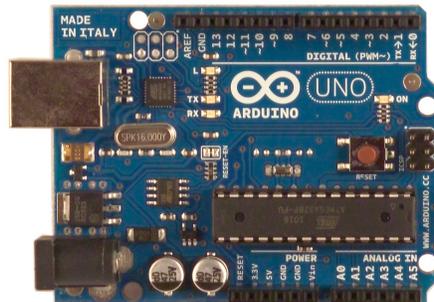


Figure 2: Arduino UNO Development Board

Figure 3 is the Digital Ad Tag Wiring schematic created using Eagle (Easy Applicable Graphical Layout Editor) version 5.10. The schematic shows all of the connections between the Arduino UNO board and the array of GLCD modules as well as the four 36 ohm resistors, four 10K ohm potentiometers and the 7408 Quad AND gate.

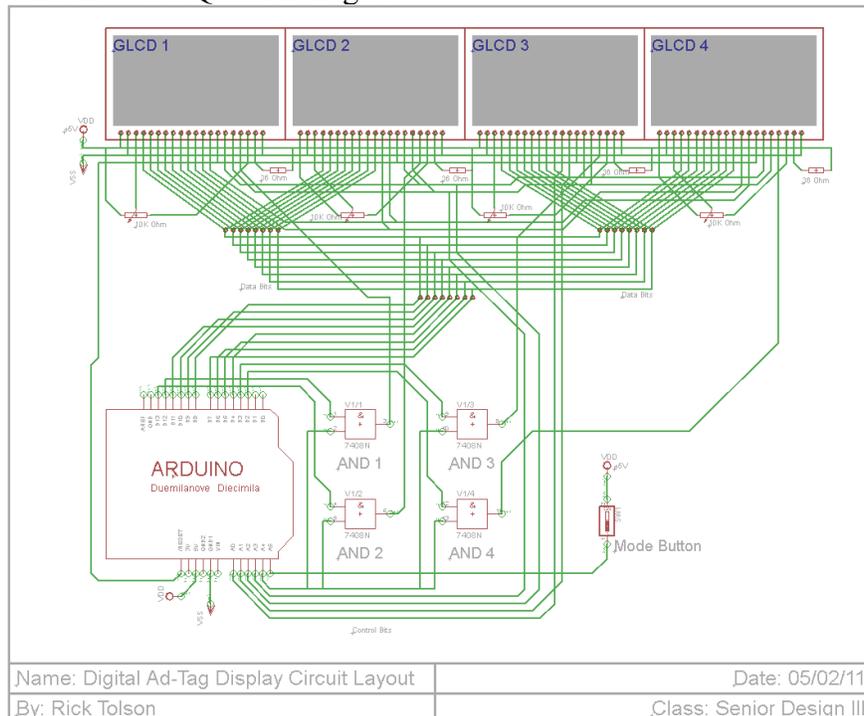


Figure 3: Digital Ad-Tag System Project Schematic

The Digital Ad-Tag System also requires the microcontroller to be connected to a laptop which would be responsible for waiting until a predetermined time and updating the code of the microcontroller to display the new weeks pricing and sale information. The laptop used in the proof of concept design is a Lenovo ThinkPad running Microsoft Windows XP. This Laptop has the Arduino development software installed and is connected to the Arduino UNO board via USB Link which also provides the 5 volt power source for the Arduino UNO and hence the array of GLCD modules. The Lenovo ThinkPad has a wireless network adapter and can be placed within the convenience store shelving unit to run silently so long as it was connected to a power source, although it will power the GLCD modules for around 3 hours on just the battery pack. The Arduino software communicates to the Arduino UNO board using a proprietary coding language that is similar to the C programming language. The code used in the microcontroller to display information on the GLCD modules begins like normal C code with #include statements calling the necessary driver header files needed to drive the displays. Then the compiler has a “void setup()” function which is where all variables are declared and ports assigned. The “void loop()” function will perform all looping operations and this is where the push button state is checked and the displays are multiplexed to receive their own display information. The Code used in the demonstration of the project is included in Appendix A of this document. These are all of the components involved in the development of the proof of concept version of the Digital Advertising Tag System. This System will Display Product Pricing and promotional sale information to the customer and allow employees to reset the shelves using digital “Plan-O-Gram” information. Figure 4 shows the flow chart used for the Digital Ad-Tag System.

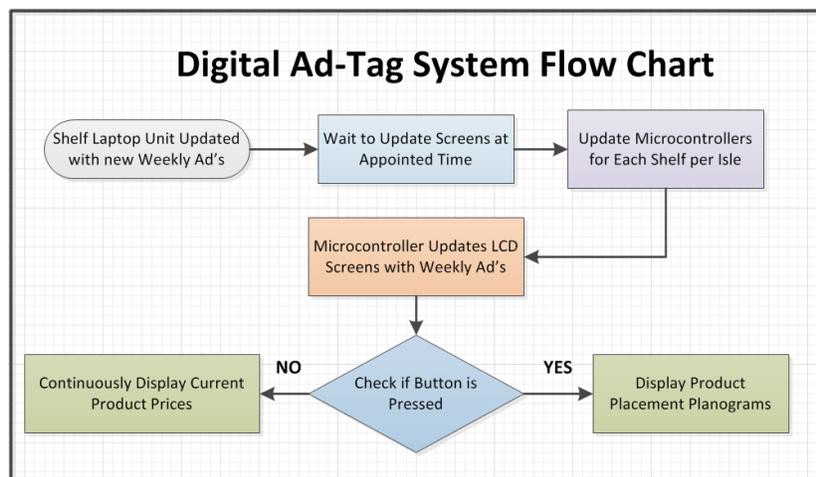


Figure 4: Digital Advertising Tag System Flow Chart.

TIMELINE

The Digital Advertising Tag System Design was started around January 1st 2011 and it initially had a fairly large scope. Figure 5 below shows the initial Timeline for completing this design by May 5th of the same year. The original design scope required the preparation of a remote display management system, and the use of passive RFID tags and RFID reader which would allow customers with a promotional savings and discount card with a built in RFID tag to approach the shelf display and see the prices change to their preferred customer price. The design also called for the creation of display management code to be written using C or C++ to keep track of time and update the displays when appropriate. As well as the creation of an Ad-Tag server which would receive updates from a store's corporate office to be passed on the the laptop and displays. This would have also required the creation of a database to store old and new Ad-Tag information. There were roadblocks and complications along the way and a decision was made to

eliminate much of the scope of this project to just show the proof of concept of displaying product pricing information and employee use “Plan-O-Gram” information. Figure 6 below displays the actual project Timeline that the project was completed on. This involved much time figuring out which microcontroller to use, this issue is explained in further detail in the Problems Encountered section of this document. The rest of the time went to preparing the laptop for use with the project, creating the wiring and connections for the Display Modules, Designing the code to show useful information on the displays, and designing the Tech Expo Poster.

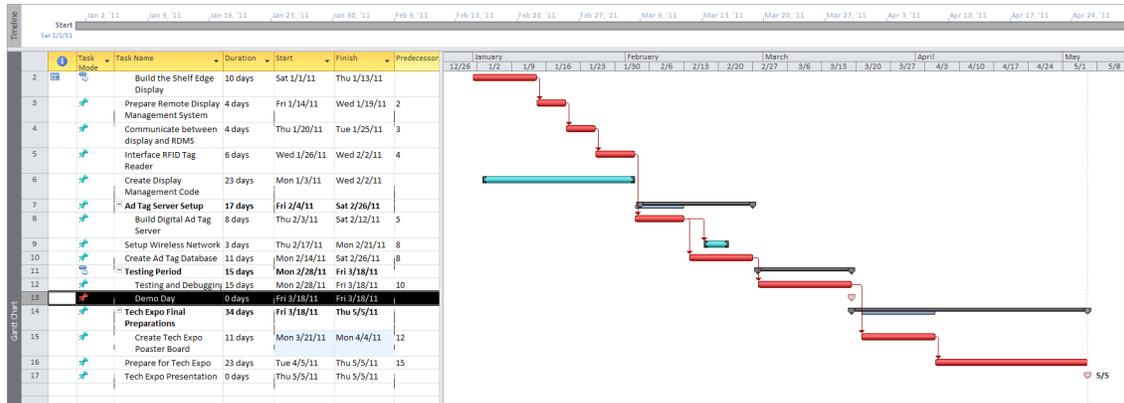


Figure 5: Initial Digital Advertising Tag System Gantt Chart.

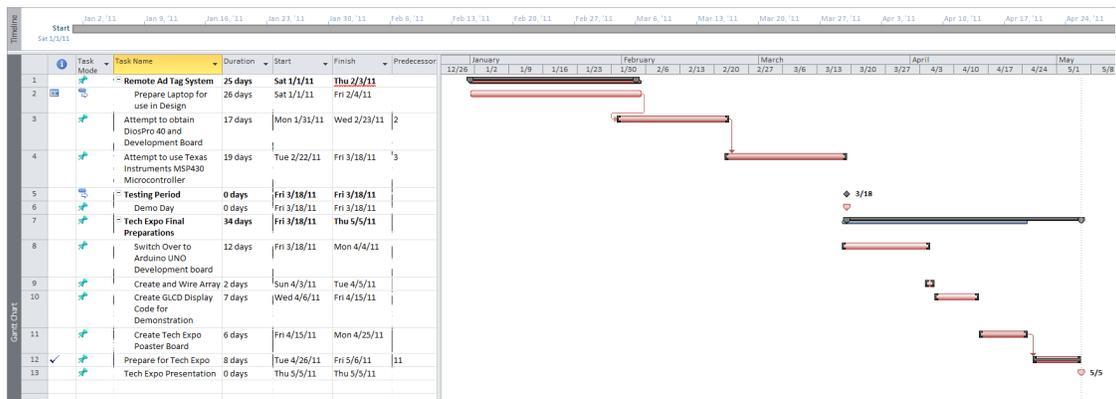


Figure 6: Final Digital Advertising Tag System Gantt Chart.

BUDGET

When the Digital Advertising Tag System was first proposed the project had a total budget cost of \$1,105.85. The current total budget for the project is now \$539.75, This cost is within a reasonable range. I was able to save much of the cost by having a suitable Ad-Tag Manager Laptop on hand. There are also many more inexpensive laptops to choose from which would reduce future budget considerations. The four GLCD Modules were purchased from www.SparkFun.com for \$19.95 each, the discount begins at 10 modules. The Arduino UNO kit was purchased from a Micro Center retail store for \$99.95, this included the Arduino UNO development board, a small breadboard, a few dozen pre-tinned wires, and handbook with examples and instructions about coding with the Arduino. I have found that just the Arduino UNO Boards themselves with the ATmega328P for around \$30.00. The Arduino UNO board, the breadboard containing 4 potentiometers, 4 resistors, an AND gate, and all the wiring are installed inside of a small black project box. This project box was purchased from a Radio Shack retail store. The wire used here had been purchased from Radio shack and any other necessary parts were purchased from Mouser Electronics at www.mouser.com.

Digital Ad-Tag Project Budget		
Parts List	Personal Cost	Actual Cost
4 LCD Displays	\$79.80	\$79.80
Ad-Tag Manager Laptop	\$0.00	\$325.00
Arduino UNO Kit	\$99.95	\$99.95
Circuit Packaging	\$10.00	\$10.00
Wire and Parts	\$25.00	\$25.00
Total	\$214.75	\$539.75

Figure 7: Digital Ad-Tag Project Budget

PROBLEMS ENCOUNTERED

The Digital Advertising Tag System was successful in proving the concept of using a shelf edge display for product pricing but there were plenty of issues along the way. Within the first month of the project I had encountered a major problem. When I purchased the GDM12864HLCM GLCD modules from SparkFun's website it was suggested to use them with the DiosPro 40 microcontroller from Kronos Robotics at www.kronosrobotics.com. The DiosPro 40 offered 64Kb of program memory and 33 I/O ports. I had placed an order with Kronos Robotics and after a month I received an email stating that they have discontinued production of that microcontroller and development board and that I should not have been able to place the order in the first place. I decided to go with another microcontroller that was a little more well known and would have more supporting documentation. I decided to use the Texas Instruments MSP430 microcontroller which has 32Kb of flash memory. After spending a couple of weeks searching the Texas Instruments website and the internet I had found that the MSP430 does not provide support for my version of Graphical LED modules, specifically Texas Instruments did not have the drivers for the KS0108 64ch segment drivers inside the module. The only option left open to me would be to attempt to write my own drivers for the MSP430 to communicate with the KS0108 segment drivers. After working with some examples I found that creating my own drivers would become a large project in itself. This had set me back a little over a month and I had decided to look for another solution. While searching for information on the KS0108 segment drivers I realized that the Arduino community has a few different software driver offerings for GLCDs using the KS0108 segment drivers. I had then decided to go with the Arduino UNO development kit and the ATmega328P microcontroller. Once the proper drivers were called in the software it was very easy to send text to the screens and create shapes such as lines, rectangles, and circles. The software drivers were supposed to support different font types and bitmap images, and after checking the driver file itself I found that there is code in place for these functions but I still could not change the font type to any of the others that I had found at the Arduino website or place bitmap images. The last major issue that occurred during the course of the digital Advertising Tag project was when I had just finished sizing, cutting, stripping, and connecting all the wires for the array of GLCD modules, when power was applied I found that one of my displays would not respond. I was concerned that the display may be broken. I checked all of the 20 pins of the display and traced them all the way back through the circuit and discovered that I had my data pins D0 – D7 on the GLCD wired backwards to the Arduino. This error was quickly corrected and all four displays were behaving as I had expected.

FUTURE ENHANCEMENTS

The Digital Ad-Tag project started with a much larger scope than it was able to achieve and there is much room for enhancement. The digital Ad-Tag system is just a proof of concept design, for this device to truly serve its purpose as a shelf edge pricing and advertising system I would want it to use full length color ePaper displays which are bright, vibrant, and viewable from many angles. I would also want to add a USB port in the shelf where a manager can plug in a USB drive containing a security key. This key would allow the plan-o-gram mode switch to be activated. This enhancement would add a level of security to the system preventing regular customers from being able to see privileged information. The original design called for the use of an RFID tag reader connected to the Display Management Laptop. Customers would then carry passive RFID tags embedded within a plastic card given to them by the store for being a member of a group. The RFID reader would detect individuals with the passive RFID tags as they approach and the display would respond by changing the currently displayed price to the discounted rate for that member. This system becomes complicated with multiple members in the same aisle and with a member standing next to a non member, the non member will see the member price until the person holding the card moves away from the RFID reader. A good enhancement idea would be to create a display control program to run on the laptop which would update the Arduino at the appropriate times and keep track of a database of old and new Ad-Tag information for every array of displays that could be attached to the system. I had run out of time by the end of the design period and I was not able to create this program. It would also be a good idea if the Digital Advertising Tag System could be tied into the Stockroom database so the employees and customers could know exactly how many of a product is on hand in the stockroom at any given time. In this way customers would not need to ask employees to check the back of the store for items as often. The possibilities for enhancement to this project are endless but the ideas suggested here are a good start.

CONCLUSION

For most of the last century grocery stores and convenience stores have become larger and more numerous. Every week all of these stores have to change Ad Tags and replace product labels which require large quantities of paper, ink, and mylar stickers. The Digital Advertising Tag System alleviate the need for convenience stores to print so many weekly labels as well as greatly reducing the occurrence of incorrectly labeled products due to human error. The proof of concept design demonstrates that a device can be created to display product pricing and weekly promotional information. The design also demonstrates how the same system can aid the employee in resetting product or creating a new product facing which is required for that week. Through completing this project I have learned many lessons about time management and how to deal with sudden design and project scope changes. I also learned how to put my tools and knowledge to work to create something that I had only envisioned just months prior. The Digital Advertising Tag System project was a great success.

REFERENCES

Arduino Main Site. Arduino team. 2006 – 2011. Accessed April 2011. <<http://arduino.cc/en/>>

Kronos Robotics and Electronics. Accessed January 2011.
<<http://www.kronosrobotics.com/xcart/home.php>>

Mouser Electronics. 1996 – 2011. Accessed March 2011. <<http://www.mouser.com/>>

Sparkfun electronics. Accessed January 2011. <<http://www.sparkfun.com/>>

Texas Instruments. 1995 – 2011. Accessed March 2011. <<http://www.ti.com/>>

Appendix A

GLCD Display Code for Arduino UNO

```

/*
 * Digital Ad-Tag Display code
 * Created by Rick Tolson
 *
 */

#include <ArduinoIcon.h>
#include <Arial14.h>
#include <ks0108.h>
#include <ks0108_Arduino.h>
#include <ks0108_Mega.h>
#include <ks0108_Panel.h>
#include <ks0108_Sanguino.h>
#include <SystemFont5x7.h>
// #include <FONT.h>

#define DISP1 12 // display 1 enable tied to pin 12
#define DISP2 13 // display 2 enable tied to pin 13
#define DISP3 3 // display 3 enable tied to pin 3
#define DISP4 2 // display 4 enable tied to pin 2
const int buttonPin = A5; // reads the push button state from analog pin 5
int buttonState = 0; // the current reading from input
int lastButtonState = LOW; // the previous reading from input
long lastDebounceTime = 0; // last time the output pin was toggled
long debounceDelay = 50; // the debounce time

void BasicDisplay(){
  /*****
   *
   * This section of code is for the basic Price Display Functions *
   *
   *****/

  /*****
   * Display 1 Data *
   *****/
  digitalWrite(DISP1, HIGH);
  digitalWrite(DISP2, LOW);
  digitalWrite(DISP3, LOW);
  digitalWrite(DISP4, LOW);

  GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
  screen
  GLCD.ClearScreen(); // clears screen
  GLCD.SelectFont(System5x7); // switch to fixed system font

  GLCD.DrawLine(0,5,125,5,BLACK);
  GLCD.DrawRect(3, 32, 38, 16, BLACK); // draws a rectangle

```

```

GLCD.CursorTo(0, 2);    // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7, WHITE);
GLCD.Puts("Cmbls Chkn Ndl"); // print given string to screen at current cursor location
GLCD.CursorTo(7,4);
GLCD.Puts("S A L E");
GLCD.CursorTo(7,5);
GLCD.Puts(" $2.35 ");
//GLCD.FillRect(54,40,40,15,BLACK);
GLCD.CursorTo(1, 5);
GLCD.SelectFont(System5x7);
GLCD.Puts("$2.50");
GLCD.CursorTo(1,7);
GLCD.Puts("655328");

/*****
 * Display 2 Data *
 *****/
digitalWrite(DISP1, LOW);
digitalWrite(DISP2, HIGH);
digitalWrite(DISP3, LOW);
digitalWrite(DISP4, LOW);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen();      // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font

GLCD.DrawLine(0,5,125,5,BLACK);
GLCD.DrawRect(3, 32, 38, 16, BLACK); // draws a rectangle

GLCD.CursorTo(0, 2);    // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7, WHITE);
GLCD.Puts("Pringles"); // print given string to screen at current cursor location
GLCD.CursorTo(7,4);
GLCD.Puts("S A L E ");
GLCD.CursorTo(7,5);
GLCD.Puts("2 for $5");
GLCD.CursorTo(1, 5);
GLCD.SelectFont(System5x7);
GLCD.Puts("$2.70");
GLCD.CursorTo(1,7);
GLCD.Puts("654673");

/*****
 * Display 3 Data *
 *****/
digitalWrite(DISP1, LOW);
digitalWrite(DISP2, LOW);
digitalWrite(DISP3, HIGH);

```

```

digitalWrite(DISPA, LOW);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen(); // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font

GLCD.DrawLine(0,5,125,5,BLACK);
GLCD.DrawRect(3, 32, 38, 16, BLACK); // draws a rectangle

GLCD.CursorTo(0, 2); // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7, WHITE);
GLCD.Puts("Canned Tuna"); // print given string to screen at current cursor location
GLCD.CursorTo(7,4);
GLCD.Puts(" S A L E ");
GLCD.CursorTo(7,5);
GLCD.Puts("5 for $10");
GLCD.CursorTo(1, 5);
GLCD.SelectFont(System5x7);
GLCD.Puts("$2.10");
GLCD.CursorTo(1,7);
GLCD.Puts("734211");

/*****
* Display 4 Data *
*****/
digitalWrite(DISPA, LOW);
digitalWrite(DISPB, LOW);
digitalWrite(DISPC, LOW);
digitalWrite(DISPD, HIGH);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen(); // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font
GLCD.DrawLine(0,5,125,5,BLACK);
GLCD.DrawRect(27, 32, 38, 16, BLACK); // draws a rectangle
GLCD.DrawVertLine(17,6,57,BLACK);

GLCD.CursorTo(5, 2); // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7, WHITE);
GLCD.Puts("Chewing Gum"); // print given string to screen at current cursor location
//GLCD.CursorTo(12,4);
//GLCD.Puts("S A L E");
//GLCD.CursorTo(7,5);
//GLCD.Puts(" $.25 ");
GLCD.CursorTo(6, 5);
GLCD.SelectFont(System5x7);
GLCD.Puts("$.25");
GLCD.CursorTo(5,7);

```

```

GLCD.Puts("655328");
}

void SecondaryMode(){
/*****
*
* This section of code is for the Plan-O-Gram Mode *
*
*****/

/*****
* Display 1 Data *
*****/
digitalWrite(DISP1, HIGH);
digitalWrite(DISP2, LOW);
digitalWrite(DISP3, LOW);
digitalWrite(DISP4, LOW);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen(); // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font

GLCD.DrawLine(0,5,125,5,BLACK);
GLCD.DrawVertLine(0,0,63,BLACK);

GLCD.CursorTo(1, 1); // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7);
GLCD.Puts("Campbells Chicken"); // print given string to screen at current cursor location
GLCD.CursorTo(1, 2);
GLCD.Puts("Noodle Soup");
GLCD.CursorTo(1,4);
GLCD.Puts("Code: 655328");
GLCD.CursorTo(2,6);
GLCD.Puts("One Facing");

/*****
* Display 2 Data *
*****/
digitalWrite(DISP1, LOW);
digitalWrite(DISP2, HIGH);
digitalWrite(DISP3, LOW);
digitalWrite(DISP4, LOW);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen(); // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font

```

```

GLCD.DrawLine(0,5,125,5,BLACK);
GLCD.DrawVertLine(0,0,63,BLACK);

GLCD.CursorTo(1, 1);    // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7);
GLCD.Puts("Pringles"); // print given string to screen at current cursor location
GLCD.CursorTo(1, 2);
GLCD.Puts("Assorted Flavors");
GLCD.CursorTo(1,4);
GLCD.Puts("Code: 654673");
GLCD.CursorTo(2,6);
GLCD.Puts("One Facing");

/*****
 * Display 3 Data *
 *****/
digitalWrite(DISP1, LOW);
digitalWrite(DISP2, LOW);
digitalWrite(DISP3, HIGH);
digitalWrite(DISP4, LOW);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen();      // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font

GLCD.DrawLine(0,5,125,5,BLACK);
GLCD.DrawVertLine(0,0,63,BLACK);

GLCD.CursorTo(1, 1);    // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7);
GLCD.Puts("StarKist Tuna"); // print given string to screen at current cursor location
GLCD.CursorTo(1,4);
GLCD.Puts("Code: 734211");
GLCD.CursorTo(2,6);
GLCD.Puts("One Facing");

/*****
 * Display 4 Data *
 *****/
digitalWrite(DISP1, LOW);
digitalWrite(DISP2, LOW);
digitalWrite(DISP3, LOW);
digitalWrite(DISP4, HIGH);

GLCD.Init(NON_INVERTED); // initializes the library, non inverted writes pixels onto a clear
screen
GLCD.ClearScreen();      // clears screen
GLCD.SelectFont(System5x7); // switch to fixed system font
GLCD.DrawLine(0,5,125,5,BLACK);

```

```

GLCD.DrawVertLine(17,0,57,BLACK);

GLCD.CursorTo(4, 1);    // 0 based coordinates for fixed width fonts
GLCD.SelectFont(System5x7);
GLCD.Puts("Big Red"); // print given string to screen at current cursor location
GLCD.CursorTo(4, 2);
GLCD.Puts("Chewing Gum");
GLCD.CursorTo(4,4);
GLCD.Puts("Code: 655328");
GLCD.CursorTo(5,6);
GLCD.Puts("One Facing");
}

```

```

void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(DISP1, OUTPUT);
  pinMode(DISP2, OUTPUT);
  pinMode(DISP3, OUTPUT);
  pinMode(DISP4, OUTPUT);
}

```

```

void loop(){
  /*
    *** Button Debounce ***

    int reading = digitalRead(buttonPin); //read the state of the pushbutton

    if (reading != lastButtonState) {
      lastDebounceTime = millis();
    }
    if ((millis() - lastDebounceTime) > debounceDelay) {
      buttonState = reading;
    }
    lastButtonState = reading;
  */
  buttonState = digitalRead(buttonPin);
  // buttonState = LOW;
  //void BasDis;
  //void PlanDis;

  if (buttonState == LOW) {
    //BasDis =
    BasicDisplay(); // call basic display subroutine

    while(buttonState == LOW){
      // wait in loop to prevent screen flicker.
      buttonState = digitalRead(buttonPin);
    }
  }
}

```

```
}  
}  
  
else if (buttonState == HIGH) {  
  //PlanDis =  
  SecondaryMode();  
  
  while(buttonState == HIGH){  
    // wait in loop to prevent screen flicker  
    buttonState = digitalRead(buttonPin);  
  }  
}  
  
}
```

Appendix B

Component Specification Sheets

*RoHS COMPLIANT



BOURNS®

Features

- 6 mm Round / Single-Turn / Cermet Industrial / Open Frame
- Cross slot adjustment options
- Horizontal and vertical mounting styles
- Dust resistant/splash resistant covers
- PC board stand-offs and retention feature
- Front and top adjust styles
- RoHS compliant*
- For trimmer applications/processing guidelines, [click here](#)

3306 - 6 mm Round Trimpot® Trimming Potentiometer

Electrical Characteristics

Standard Resistance Range100 ohms to 1 megohm
 (see standard resistance table)
 Resistance Tolerance ±25 % std.
 (tighter tolerance available)
 Absolute Minimum Resistance 2 % max. (≤2K = 30 ohms)
 Contact Resistance Variation. 3 % max.
 Resolution Essentially infinite
 Adjustment Angle 215 ° nom.

Environmental Characteristics

Power Rating (100 volts max.)
 70 °C 0.2 watt
 Temperature Range ... -25 °C to +100 °C
 Temperature Coefficient ... ±250 ppm/°C
 Load Life ..1,000 hours 0.2 watt @ 70 °C
 (5 % ΔTR)

Physical Characteristics

Mechanical Angle 260 ° ±20 °
 Torque (Operating) 4.5 oz-in. max.
 Stop Strength 6.5 oz -in. min.
 Terminals Solderable pins
 Marking Manufacturer's trademark, resistance code
 Standard Packaging 200 pcs. per bag
 Adjustment Tool H-90

Aqueous cleaning not recommended.

How To Order

3306 W - 1 - 103

Model _____
 Style _____
 Standard Product _____
 Resistance Code _____

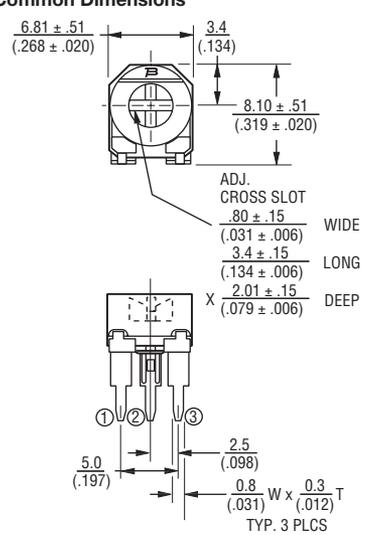
Standard Resistance Table

Resistance (Ohms)	Resistance Code
100	101
200	201
500	501
1,000	102
2,000	202
5,000	502
10,000	103
20,000	203
25,000	253
50,000	503
100,000	104
200,000	204
250,000	254
500,000	504
1,000,000	105

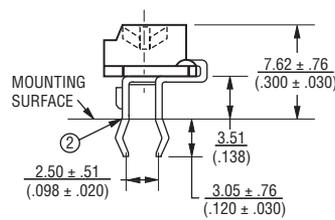
Popular distribution resistance values listed in boldface.

Product Dimensions

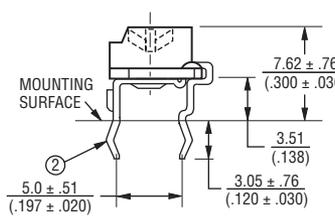
Top Adjust Common Dimensions



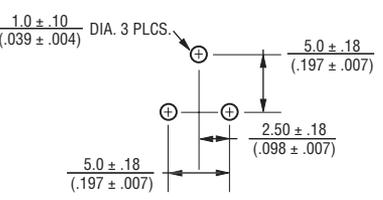
3306P



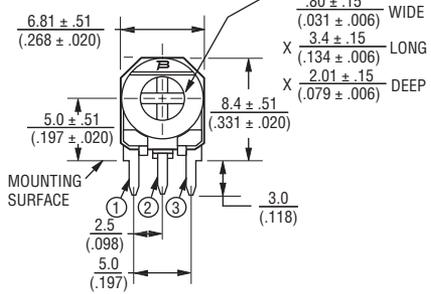
3306F



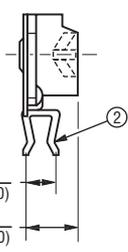
SUGGESTED PWB LAYOUT - STYLE F



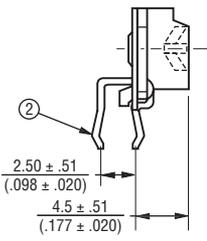
Side Adjust Common Dimensions



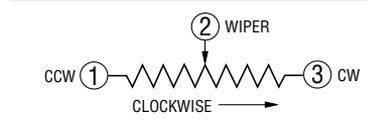
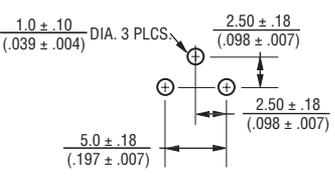
3306W



3306K



SUGGESTED PWB LAYOUT - STYLES K, P, W



DIMENSIONS: $\frac{\text{MM}}{\text{(INCHES)}}$
 TOLERANCES: ± $\frac{0.3}{(.012)}$ EXCEPT WHERE NOTED

REV. 11/10

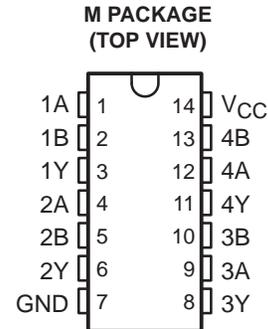
*RoHS Directive 2002/95/EC Jan 27, 2003 including Annex. "Trimpot" is a registered trademark of Bourns, Inc. Specifications are subject to change without notice. Customers should verify actual device performance in their specific applications.

CD74HC08-EP QUADRUPLE 2-INPUT POSITIVE-AND GATES

SCLS477B – APRIL 2003 – REVISED APRIL 2004

- **Controlled Baseline**
 - One Assembly/Test Site, One Fabrication Site
- **Extended Temperature Performance of Up To -55°C to 125°C**
- **Enhanced Diminishing Manufacturing Sources (DMS) Support**
- **Enhanced Product-Change Notification**
- **Qualification Pedigree†**
- **Buffered Inputs**
- **Typical Propagation Delay 7 ns at $V_{CC} = 5\text{ V}$, $C_L = 15\text{ pF}$, $T_A = 25^{\circ}\text{C}$**
- **Fanout (Over Temperature Range)**
 - Standard Outputs . . . 10 LSTTL Loads
 - Bus Driver Outputs . . . 15 LSTTL Loads
- **Balanced Propagation Delay and Transition Times**
- **Significant Power Reduction Compared to LSTTL Logic ICs**
- **2-V to 6-V V_{CC} Operation**
- **High Noise Immunity N_{IL} or $N_{IH} = 30\%$ of V_{CC} at $V_{CC} = 5\text{ V}$**
- **CMOS Input Compatibility, $I_I \leq 1\ \mu\text{A}$ at V_{OL} , V_{OH}**

† Component qualification in accordance with JEDEC and industry standards to ensure reliable operation over an extended temperature range. This includes, but is not limited to, Highly Accelerated Stress Test (HAST) or biased 85/85, temperature cycle, autoclave or unbiased HAST, electromigration, bond intermetallic life, and mold compound life. Such qualification testing should not be viewed as justifying use of this component beyond specified performance and environmental limits.



description/ordering information

The CD74HC08 logic gates utilize silicon-gate CMOS technology to achieve operating speeds similar to LSTTL gates, with the low power consumption of standard CMOS integrated circuits. All devices can drive 10 LSTTL loads.

ORDERING INFORMATION

T_A	PACKAGE‡		ORDERABLE PART NUMBER	TOP-SIDE MARKING
-40°C to 125°C	SOIC – M	Tape and reel	CD74HC08QM96EP	HC08QEP
-55°C to 125°C	SOIC – M	Tape and reel	CD74HC08MM96EP§	HC08MEP

‡ Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

§ Product Preview

FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	H	H
L	X	L
X	L	L



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

CD74HC08-EP

QUADRUPLE 2-INPUT POSITIVE-AND GATES

SCLS477B – APRIL 2003 – REVISED APRIL 2004

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC}	-0.5 V to 7 V
Input clamp current, I_{IK} ($V_I < -0.5$ V or $V_I > V_{CC} + 0.5$ V) (see Note 1)	± 20 mA
Output clamp current, I_{OK} ($V_O < -0.5$ V or $V_O > V_{CC} + 0.5$ V) (see Note 1)	± 20 mA
Continuous output current, I_O ($V_O > -0.5$ or $V_O < V_{CC} + 0.5$ V)	± 25 mA
Continuous current through V_{CC} or GND	± 50 mA
Package thermal impedance, θ_{JA} (see Note 2)	180°C/W
Maximum junction temperature, T_J	150°C
Lead temperature (during soldering):	
At distance $1/16 \pm 1/32$ inch ($1,59 \pm 0,79$ mm) from case for 10 s max	300°C
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
 2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions (see Note 3)

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	2	5	6	V
V_{IH}	High-level input voltage	$V_{CC} = 2$ V	1.5		V
		$V_{CC} = 4.5$ V	3.15		
		$V_{CC} = 6$ V	4.2		
V_{IL}	Low-level input voltage	$V_{CC} = 2$ V		0.5	V
		$V_{CC} = 4.5$ V		1.35	
		$V_{CC} = 6$ V		1.8	
V_I	Input voltage	0		V_{CC}	V
V_O	Output voltage	0		V_{CC}	V
$\Delta t/\Delta v$	Input transition rise/fall time	$V_{CC} = 2$ V		1000	ns
		$V_{CC} = 4.5$ V		500	
		$V_{CC} = 6$ V		400	
T_A	Operating free-air temperature	Q suffix	-40	125	°C
		M suffix	-55	125	

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.



CD74HC08-EP QUADRUPLE 2-INPUT POSITIVE-AND GATES

SCLS477B – APRIL 2003 – REVISED APRIL 2004

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS		I _O (mA)	V _{CC}	T _A = 25°C			MIN	MAX	UNIT
					MIN	TYP	MAX			
V _{OH}	V _I = V _{IH} or V _{IL}	CMOS loads	-0.02	2 V	1.9			1.9	V	
			-0.02	4.5 V	4.4			4.4		
			-0.02	6 V	5.9			5.9		
		TTL loads	-4	4.5 V	3.98			3.7		
			-5.2	6 V	5.48			5.2		
V _{OL}	V _I = V _{IH} or V _{IL}	CMOS loads	0.02	2 V	0.1			0.1	V	
			0.02	4.5 V	0.1			0.1		
			0.02	6 V	0.1			0.1		
		TTL loads	4	4.5 V	0.26			0.4		
			5.2	6 V	0.26			0.4		
I _I	V _I = V _{CC} or GND			6 V	±0.1			±1	μA	
I _{CC}	V _I = V _{CC} or GND		0	6 V	2			40	μA	
C _i					10			10	pF	

switching characteristics over recommended operating free-air temperature range (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	CONDITIONS	V _{CC}	T _A = 25°C			MIN	MAX	UNIT
					MIN	TYP	MAX			
t _{pd}	A or B	Y	C _L = 50 pF	2 V	90			135	ns	
				4.5 V	18			27		
				6 V	15			23		
			C _L = 15 pF	5 V	7					
t _t	A or B	Y	C _L = 50 pF	2 V	75			110	ns	
				4.5 V	15			22		
				6 V	13			19		

operating characteristics, T_A = 25°C, V_{CC} = 5V

PARAMETER	TEST CONDITIONS	TYP	UNIT
C _{pd} Power dissipation capacitance per gate (see Note 4)	No load	37	pF

NOTE 4: C_{pd} is used to determine the dynamic power consumption, per gate.

$$P_D = V_{CC}^2 f_I (C_{pd} + C_L)$$

f_I = input frequency

C_L = output load capacitance

V_{CC} = supply voltage



User's Guide

GDM12864HLCM

(Liquid Crystal Display Module)

For product support, contact

XIAMEN OCULAR OPTICS CO.,LTD.

厦门高卓立光电有限公司

South2/F.,Guangxia Building Torch Hi-tech
Industrial Development Area Xiamen, China

36100 中国厦门火炬高技术产业开发区光厦楼南二楼

Tel: 86-592-5650516 Fax: 86-592-5650695

CONTENTS

Mechanical diagram

Absolute maximum ratings

Interface pin connections

Optical characteristics

Electrical characteristics

KS0107B

KS0108B

Write or read cycle

Timing characteristics

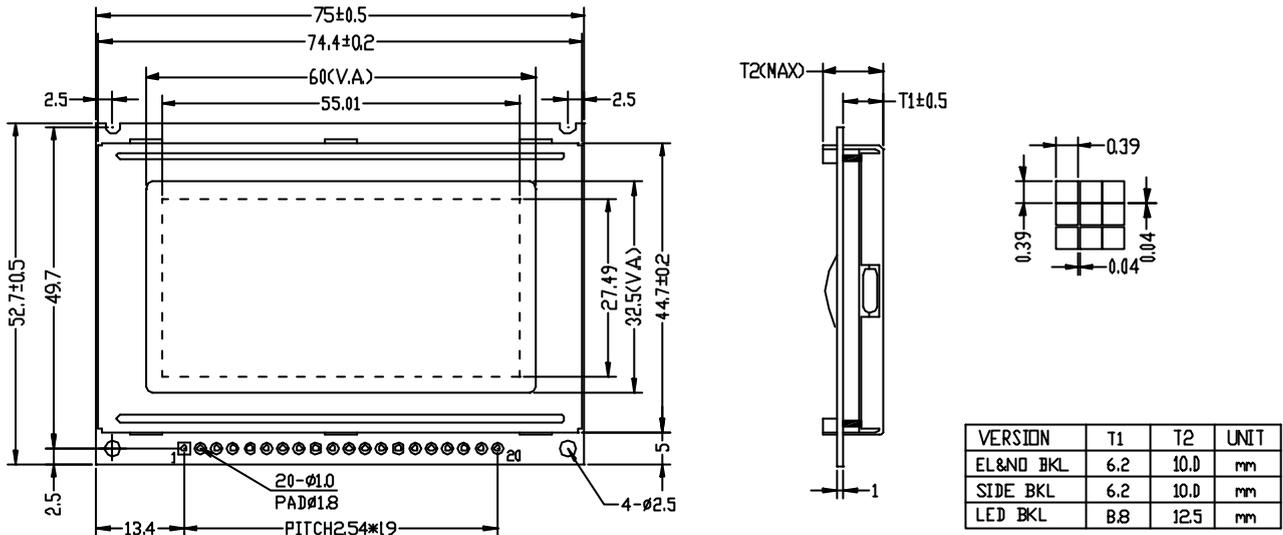
Block diagram

Display commands

Reliability and lift time

Operating Principles & Methods

➤ Mechanical diagram



➤ Absolute maximum ratings

Item	Symbol	Min.	Max.	Unit
Supply voltage for logic	Vdd - Vss	0	6.5	V
Input voltage	Vin	0	Vdd	
Operating temperature range	T0p	-20	70	℃
Storage temperature range	Tst	-25	75	

➤ Interface pin connections

Pin No.	Symbol	Level	Description
1	Vdd	5.0V	Supply voltage for logic and LCD (+)
2	Vss	0V	Ground
3	V0	-	Operating voltage for LCD (variable)
4~11	DB0~DB7	H/L	Data bit 0~7
12	CS2	L	Chip select signal for IC2
13	CS1	L	Chip select signal for IC1
14	/RES	L	Reset signal
15	R/W	H/L	H: read (MUP← module),L: write (MPU→ module)
16	D/I	H/L	H: data, L: instruction code
17	E	H, H L	Chip enable signal
18	VEE	-	Operating voltage for LCD (variable)
19	A	4.2V	Backlight power supply
20	K	0V	Backlight power supply

XIAMEN OCULAR OPTICS CO., LTD.

3

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

➤ Optical characteristics

STN Type display module (Ta=25°C, Vdd=5.0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Viewing angle	θ	$Cr \geq 2$	-60	-	35	deg
			-40	-	40	
Contrast ratio (rise)	Cr		-	6	-	
Response time (fall)	Tr	-	-	150	250	ms
	Tr	-	-	150	250	ms

➤ Electrical characteristics

Item	Symbol	Condition	Standard value			Unit	
			Min.	Typ.	Max.		
Supply voltage for	Logic	Vdd - Vss	-	4.75	5.0	5.25	V
	LCD	Vdd-V0	-	-	9.5	-	
Supply current for	Logic	Idd	-	-	2.5	-	mA
	LCD	Iee	-	-	1.0	-	
Operating voltage for LCD (Recommended)	Vdd-v0	-	-	-	-	V	
		25°C	-	9.5	-		
		-	-	-	-		
Input voltage	H: level	Vih	High level	0.7Vdd	-	Vdd	V
	L: Level	Vil	Low level	0	-	0.3Vdd	

Electrical Absolute Maximum Ratings (KS0107B)

Parameter	Symbol	Rating	Unit	Note
Operating voltage	V _{DD}	-0.3 ~ +7.0	V	*1
Supply voltage	V _{EE}	V _{DD} -19.0 ~ V _{DD} +0.3	V	*4
Driver supply voltage	V _B	-0.3 ~ V _{DD} +0.3	V	*1,2
	V _{LCD}	V _{EE} -0.3 ~ V _{DD} +0.3	V	*3,4

*Notes:

- *1. Based on V_{SS} = 0V
- *2. Applies to input terminals and I/O terminals at high impedance. (Except V0L, V1L, V4L, and V5L)
- *3. Applies to V0L, V1L, V4L, and V5L.
- *4. Voltage level: V_{DD} ≥ V0 ≥ V1 ≥ V2 ≥ V3 ≥ V4 ≥ V5 ≥ V_{EE}

DC Electrical Characteristics (KS0107B)

(VDD= 4.5 to 5.5V, VSS=0V, VDD-VEE=8~17V, Ta= -30 to +85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Operating voltage	V _{DD}	-	4.5	-	5.5	V	
Input voltage	V _{IH}	-	0.7V _{DD}	-	V _{DD}		*1
	V _{IL}	-	V _{SS}	-	0.3V _{DD}		
output voltage	V _{OH}	I _{OH} = -0.4mA	V _{DD} -0.4	-	-		*2
	V _{OL}	I _{OL} = 0.4mA	-	-	0.4		
Input leakage current	I _{LKG}	V _{IN} = V _{DD} ~ V _{SS}	-1.0	-	+1.0	μA	*1
OSC Frequency	f _{osc}	Rf=47kΩ ±2% Cf=20pF ±5%	315	450	585	kHz	
On Resistance (Vdiv-Ci)	R _{ONS}	V _{DD} -V _{EE} =17V Load current ±150μA	-	-	1.5	kΩ	
Operating current	I _{DD1}	Master mode 1/128 Duty	-	-	1.0	mA	*3
	I _{DD2}	Master mode 1/128 Duty	-	-	0.2		*4
Supply Current	I _{EE}	Master mode 1/128 Duty	-	-	0.1		*5
Operating Frequency	f _{op1}	Master mode External Duty	50	-	600	kHz	
	f _{op2}	Slave mode	0.5	-	1500		

Notes

- *1. Applies to input terminals FS, DS1, DS2, CR, SHL, MS and PCLK2 and I/O terminals DIO1, DIO2, M, and CL2 in the input state.
- *2. Applies to output terminals CLK1, CLK2 and FRM and I/O terminals DIO1, DIO2, M, and CL2 in the output state.
- *3. This value is specified about current flowing through V_{ss}.
Internal oscillation circuit: Rf=47kΩ, cf=20pF
Each terminals of DS1, DS2, FS, SHL, and MS is connected to V_{DD} and out is no load.
- *4. This value is specified about current flowing through V_{ss}.
Each terminals is DS1, DS2, FS, SHL, PCLK2 and CR is connected to V_{DD}. MS is connected to V_{ss} and CL2, M, DIO1 is external clock.
- *5. This value is specified about current flowing through V_{EE}. Don't connect to V_{LCD} (V1~V5).

Electrical Absolute Maximum Ratings (KS0108B)

Parameter	Symbol	Rating	Unit	Note
Operating voltage	V_{DD}	-0.3 ~ +7.0	V	*1
Supply voltage	V_{EE}	$V_{DD}-19.0 \sim V_{DD}+0.3$	V	*4
Driver supply voltage	V_B	-0.3 ~ $V_{DD}+0.3$	V	*1,3
	V_{LCD}	$V_{EE}-0.3 \sim V_{DD}+0.3$	V	*2

*Notes:

- *1. Based on $V_{SS} = 0V$
- *2. Applies the same supply voltage to V_{EE} . $V_{LCD}=V_{DD}-V_{EE}$.
- *3. Applies to M, FRM, CLK1, CLK2, CL, RESETB, ADC, CS1B, CS2B, CS3, E, R/W, RS and DB0~DB7.
- *4. Applies V0L, V2L, V3L and V5L.

Voltage level: $V_{DD} \geq V0 \geq V1 \geq V2 \geq V3 \geq V4 \geq V5 \geq V_{EE}$

DC Electrical Characteristics (KS0108B)

($V_{DD}= 4.5$ to $5.5V$, $V_{SS}=0V$, $V_{DD}-V_{EE}=8\sim 17V$, $T_a= -30$ to $+85^{\circ}C$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Operating voltage	V_{DD}	-	4.5	-	5.5	V	
Input High voltage	V_{IH1}	-	$0.7V_{DD}$	-	V_{DD}		*1
	V_{IH2}	-	2.0	-	V_{DD}		*2
Input Low voltage	V_{IL1}	-	0	-	$0.3V_{DD}$		*1
	V_{IL2}	-	0	-	0.8		*2
Output High Voltage	V_{OH}	$I_{OH}= -0.2mA$	2.4	-	-		*3
Output Low Voltage	V_{OL}	$I_{OL}= 1.6mA$	-	-	0.4	*3	
Input leakage current	I_{LKG}	$V_{IN}= V_{SS} \sim V_{DD}$	-1.0	-	+1.0	μA	*4
Three-state (OFF) Input Current	I_{TSL}	$V_{IN}= V_{SS} \sim V_{DD}$	-5.0	-	5.0		*5
Driver Input leakage current	I_{DIL}	$V_{IN}= V_{EE} \sim V_{DD}$	-2.0		2.0		*6
On Resistance (Vdiv-Ci)	R_{ONS}	$V_{DD}-V_{EE}=15V$ Load current $\pm 100\mu A$	-	-	7.5	$k\Omega$	*8
Operating current	I_{DD1}	During Display	-	-	0.1	mA	*7
	I_{DD2}	During Access Access Cycle=1MHz	-	-	0.5		*7

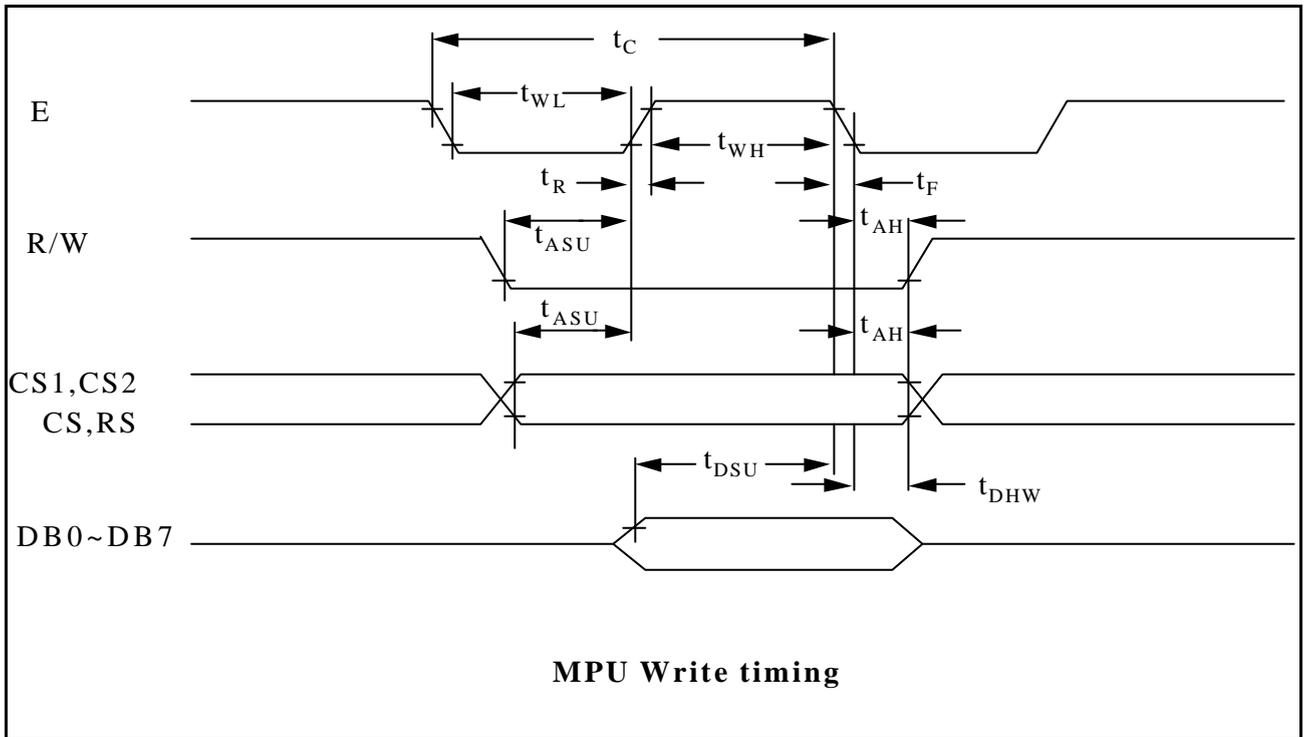
Notes

- *1. CL, FRM, M, RSTB, CLK1, CLK2
- *2. CS1B, CS2B, CS3, E, R/W, RS, DB0~DB7
- *3. DB0~DB7
- *4. Except DB0~DB7
- *5. DB0~DB7 at high impedance
- *6. V0, V1, V3, V3, V4, V5
- *7. 1/64 duty, FCLK=250KHZ, Frame Frequency=70HKZ, Output: No Load
- *8. $V_{DD}-V_{EE}=15.5V$
 $V0L>V2L \geq V_{DD}-2/7(V_{DD}-V_{EE}) > V3L = V_{EE}+2/7(V_{DD}-V_{EE}) > V5L$

➤ Write or read cycle

Characteristic	Symbol	Min.	Typ.	Max.	Unit
E cycle	Tc	1000	-	-	ns
E high level width	Twh	450	-	-	ns
E low level width	Twl	450	-	-	ns
E rise time	Tr	-	-	25	ns
E fall time	Tf	-	-	25	ns
Address set-up time	Tasu	140	-	-	ns
Address hold time	Tah	10	-	-	ns
Data set-up time	Tdsu	200	-	-	ns
Data delay time	Td	-	-	320	ns
Data hold time (write)	Tdhw	10	-	-	ns
Data hold time (read)	Tdhr	20	-	-	ns

✧ Write timing

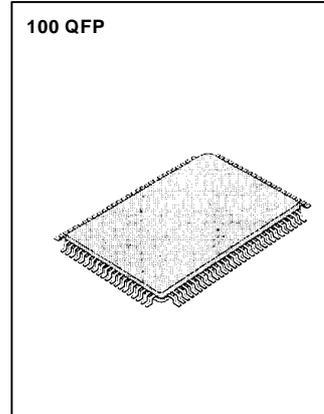


KS0108B

64CH SEGMENT DRIVER FOR DOT MATRIX LCD

INTRODUCTION

The KS0108B is a LCD driver LSI with 64 channel output for dot matrix liquid crystal graphic display system. This device consists of the display RAM, 64 bit data latch 64 bit drivers and decoder logics. It has the internal display RAM for storing the display data transferred from a 8 bit micro controller and generates the dot matrix liquid crystal driving signals corresponding to stored data. The KS0108B composed of the liquid crystal display system in combination with the KS0107B (64 common driver)



FEATURES

- Dot matrix LCD segment driver with 64 channel output
- Input and Output signal
 - Input: 8 bit parallel display data
 - Control signal from MPU
 - Splitted bias voltage (V1R, V1L, V2R, V2L, V3R, V3L, V4R, V4L)
 - Output: 64 channel waveform for LCD driving.
- Display data is stored in display data RAM from MPU.
- Interface RAM
 - Capacity: 512 bytes (4096 bits)
 - RAM bit data: RAM bit data = 1:ON
RAM bit data = 0:OFF
- Applicable LCD duty: 1/32~1/64
- LCD driving voltage: 8V~17V($V_{DD}-V_{EE}$)
- Power supply voltage: + 5V±10%

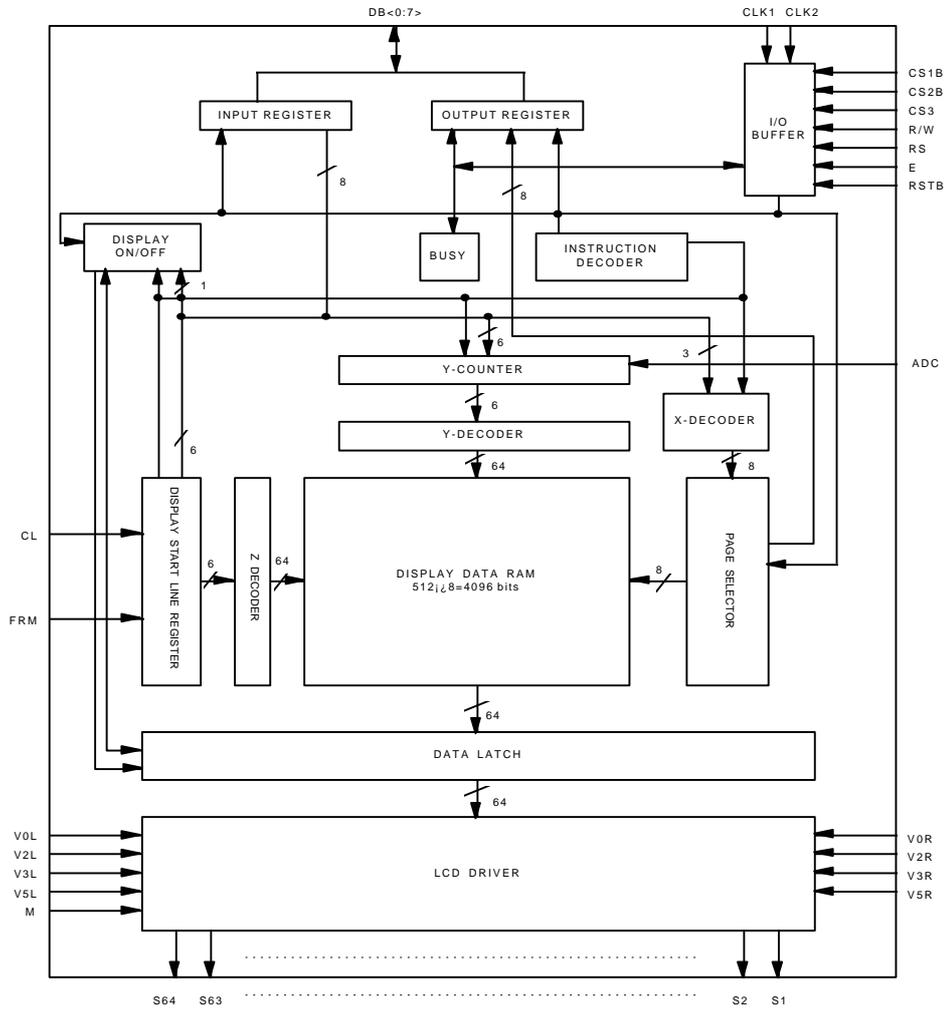
Driver		Controller
COMMON	SEGMENT	
KS0107B	Other KS0108B	MPU

- High voltage CMOS process.
- 100QFP and bare chip available.

KS0108B

64CH SEGMENT DRIVER FOR DOT MATRIX LCD

BLOCK DIAGRAM



KS0108B

64CH SEGMENT DRIVER FOR DOT MATRIX LCD

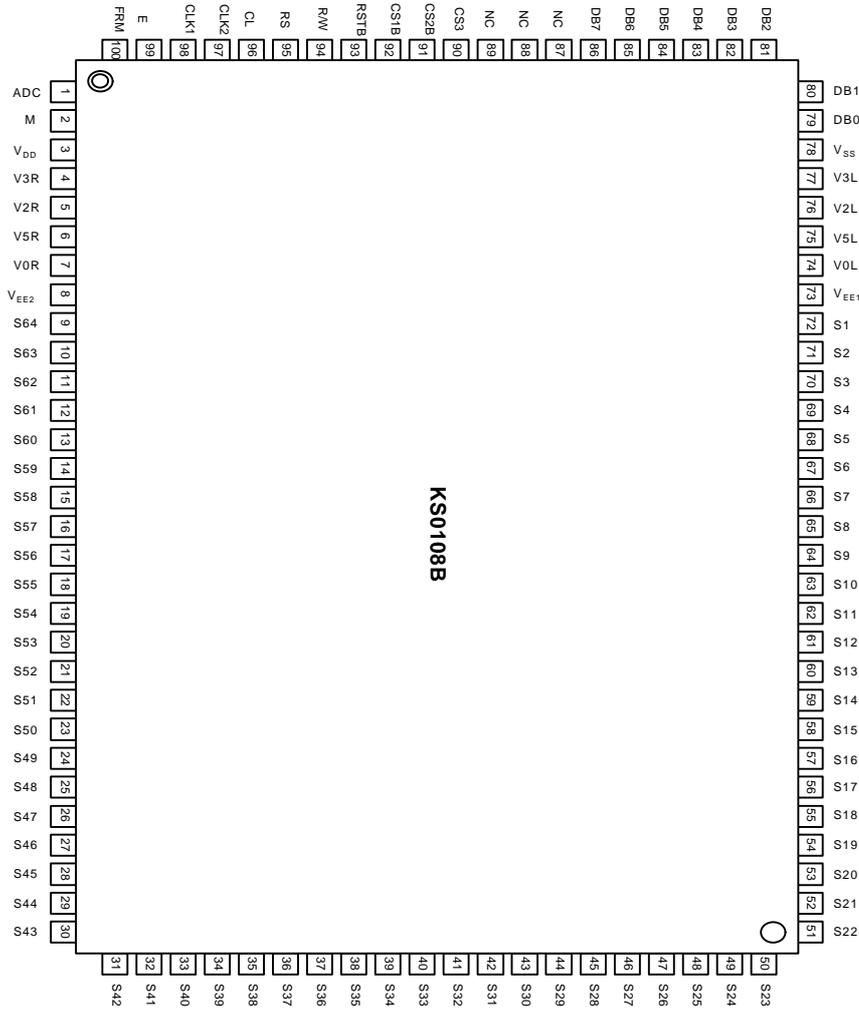


Fig.2. 100QFP Top View

PIN DESCRIPTION

PIN (NO)	SYMBOL	INPUT/OUTPUT	DESCRIPTION				
3 78 73, 8	V_{DD} V_{SS} $V_{EE1,2}$	Power	For internal logic circuit (+5V±10%) GND (0V) For LCD driver circuit $V_{SS}=0V$, $V_{DD}=5V$;±10% $V_{DD}-V_{EE}=8V\sim 17V$ V_{EE1} and V_{EE2} is connected by the same voltage.				
74, 7 76, 5 77, 4 75, 6	V_{0L} , V_{0R} V_{2L} , V_{2R} V_{3L} , V_{3R} V_{5L} , V_{5R}	Power	Bias supply voltage terminals to drive the LCD. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Select Level</td> <td>Non-Select Level</td> </tr> <tr> <td>$V_{0L(R)}$, $V_{5L(R)}$</td> <td>$V_{2L(R)}$, $V_{3L(R)}$</td> </tr> </table>	Select Level	Non-Select Level	$V_{0L(R)}$, $V_{5L(R)}$	$V_{2L(R)}$, $V_{3L(R)}$
Select Level	Non-Select Level						
$V_{0L(R)}$, $V_{5L(R)}$	$V_{2L(R)}$, $V_{3L(R)}$						
92 91 90	CS1B CS2B CS3	Input	Chip selection In order to interface data for input or output The terminals have to be CS1B=L, CS2B=L, and CS3=H.				
2	M	Input	Alternating signal input for LCD driving.				
1	ADC	Input	Address control signal of Y address counter. ADC=H→DB<0:7>=0→Y0→S1 DB<0:7>=63→Y63→S64 ADC=L→DB<0:7>=0→Y63→S64 DB<0:7>=63→Y0→S1				
100	FRM	Input	Synchronous control signal. Presets the 6-bit Z counter and synchronizes the common signal with the frame signal when the frame signal becomes high.				
99	E	Input	Enable signal. write mode (R/W=L) → data of DB<0:7> is latched at the falling edge of E. read mode (R/W=H) → DB<0:7> appears the reading data while E is at high level.				
98 97	CLK1 CLK2	Input	2 phase clock signal for internal operation. Used to execute operations for input/output of display RAM data and others.				
96	CL	Input	Display synchronous signal. Display data is latched at rising time of the CL signal and increments the Z-address counter at the CL falling time.				
95	RS	Input	Data or Instruction. RS=H→DB<0:7> : Display RAM Data RS=L→DB<0:7> : Instruction Data				
94	R/W	Input	Read or Write. R/W=H → Data appears at DB<0:7> and can be read by the CPU while E=H, CS1B=L, CS2B=L and CS3=H. R/W=L; Display data DB<0:7> can be written at falling of E when CS1B=L, CS2B=L and CS3=H.				
79~86	DB0~DB7	Input/Output	Data bus. These state I/O common terminal.				

KS0108B

64CH SEGMENT DRIVER FOR DOT MATRIX LCD

PIN DESCRIPTION(continued)

PIN (NO)	NAME	INPUT/OUTPUT	DESCRIPTION													
72~9	S1~S64	Output	LCD Segment driver output. Display RAM data 1:ON Display RAM data 0:OFF (Relation of display RAM data & M) <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>M</th> <th>DATA</th> <th>Output Level</th> </tr> </thead> <tbody> <tr> <td rowspan="2">L</td> <td>L</td> <td>V_2</td> </tr> <tr> <td>H</td> <td>V_0</td> </tr> <tr> <td rowspan="2">H</td> <td>L</td> <td>V_3</td> </tr> <tr> <td>H</td> <td>V_5</td> </tr> </tbody> </table>	M	DATA	Output Level	L	L	V_2	H	V_0	H	L	V_3	H	V_5
M	DATA	Output Level														
L	L	V_2														
	H	V_0														
H	L	V_3														
	H	V_5														
93	RSTB	Input	Reset signal. When RSTB=L, (1) ON/OFF register becomes set by 0. (display off) (2) Display start line register becomes set by 0 (Z-address 0 set, display from line 0) After releasing reset, this condition can be changed only by instruction.													
87~89	NC		No connection.(open)													

MAXIMUM ABSOLUTE LIMIT

Characteristic	Symbol	Value	Unit	Note
Operating Voltage	V_{DD}	-0.3~+7.0	V	*1
Supply Voltage	V_{EE}	$V_{DD}-19.0-V_{DD}+0.3$	V	*4
Driver Supply Voltage	V_B	-0.3~ $V_{DD}+0.3$	V	*1,3
	V_{LCD}	$V_{EE}-0.3-V_{DD}+0.3$	V	*2
Operating Temperature	T_{OPR}	-30~+85	°C	
Storage Temperature	T_{STG}	-55~+125	°C	

*1. Based on $V_{SS}=0V$.

*2. Applies the same supply voltage to V_{EE1} and V_{EE2} . $V_{LCD}=V_{DD}-V_{EE}$.

*3. Applies to M, FRM, CL, RSTB, ADC, CLK1, CLK2, CS1B, CS2B, CS3, E, R/W, RS and DB0~DB7.

*4. Applies V0L(R), V2L(R), V3L(R) and V5L(R).

Voltage level: $V_{DD} \geq V_{0L} = V_{0R} \geq V_{2L} = V_{2R} \geq V_{3L} = V_{3R} \geq V_{5L} = V_{5R} \geq V_{EE}$.



ELECTRICAL CHARACTERISTICSDC Characteristics ($V_{DD}=4.5\sim 5.5V$, $V_{SS}=0V$, $V_{DD}-V_{EE}=8\sim 17V$, $T_a=-30\sim +85^\circ C$)

Characteristic	Symbol	Condition	Min	Typ	Max	Unit	Note
Input High Voltage	V_{IH1}	-	$0.7V_{DD}$	-	V_{DD}	V	*1
	V_{IH2}	-	2.0	-	V_{DD}	V	*2
Input Low Voltage	V_{IL1}	-	0	-	$0.3V_{DD}$	V	*1
	V_{IL2}	-	0	-	0.8	V	*2
Output High Voltage	V_{OH}	$I_{OH}=-200\mu A$	2.4	-	-	V	*3
Output Low Voltage	V_{OL}	$I_{OL}=1.6mA$	-	-	0.4	V	*3
Input Leakage Current	I_{LKG}	$V_{IN}=V_{SS}\sim V_{DD}$	-1.0	-	1.0	μA	*4
Three-state(OFF) Input Current	I_{TSL}	$V_{IN}=V_{SS}\sim V_{DD}$	-5.0	-	5.0	μA	*5
Driver Input Leakage Current	I_{DIL}	$V_{IN}=V_{EE}\sim V_{DD}$	-2.0	-	2.0	μA	*6
Operating Current	I_{DD1}	During Display	-	-	100	μA	*7
	I_{DD2}	During Access Access Cycle=1MHz	-	-	500	μA	*7
On Resistance	R_{ON}	$V_{DD}-V_{EE}=15V$ $I_{LOAD}=0.1mA$	-	-	7.5	$K\Omega$	*8

- *1. CL, FRM, M, RSTB, CLK1, CLK2
2. CS1B, CS2B, CS3, E, R/W, RS, DB0-DB7
3. DB0-DB7
4. Excepted DB0-DB7
5. DB0-DB7 at High Impedance
6. V0L(R), V2L(R), V3L(R), V5L(R)
7. 1/64 duty, FCLK=250KHZ, Frame Frequency=70HZ, Output: No Load
8. $V_{DD}-V_{EE}=15.5V$
 $V0L(R)>V2L(R)=V_{DD}-2/7$ ($V_{DD}-V_{EE}$) $>V3L(R)=V_{EE}+2/7$ ($V_{DD}-V_{EE}$) $>V5L(R)$

AC Characteristics ($V_{DD}=5V\pm 10\%$, $V_{SS}=0V$, $T_a=-30^\circ C\sim +85^\circ C$)

(1) Clock Timing

Characteristic	Symbol	Min	Typ	Max	Unit
CLK1, CLK2 Cycle Time	t_{CY}	2.5	-	20	μS
CLK1 'LOW' Level Width	t_{WL1}	625	-	-	ns
CLK2 'LOW' Level Width	t_{WL2}	625	-	-	
CLK1 'HIGH' Level Width	t_{WH1}	1875	-	-	
CLK2 'HIGH' Level Width	t_{WH2}	1875	-	-	
CLK1-CLK2 Phase Difference	t_{D12}	625	-	-	
CLK2-CLK1 Phase Difference	t_{D21}	625	-	-	
CLK1, CLK2 Rise Time	t_R	-	-	150	
CLK1, CLK2 Fall Time	t_F	-	-	150	

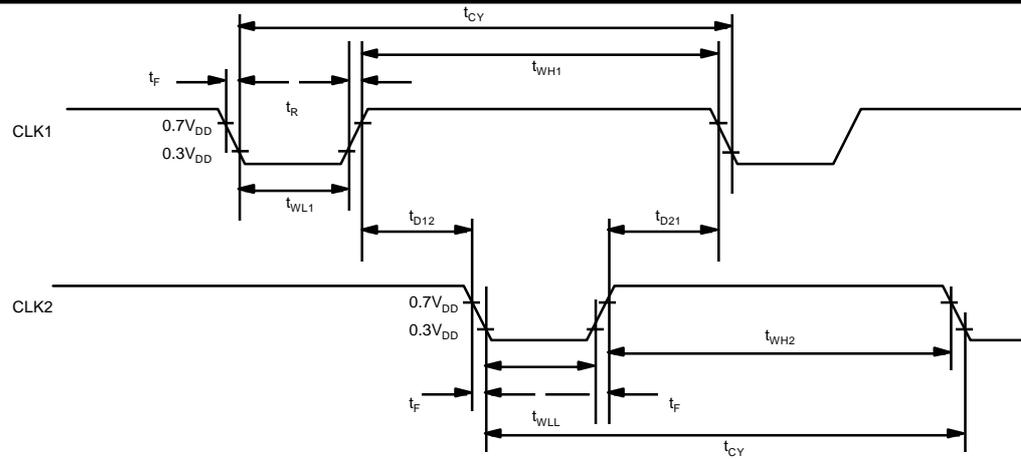


Fig 1. External clock waveform

(2) Display Control Timing

Characteristic	Symbol	Min	Typ	Max	Unit
FRM Delay Time	t_{DF}	-2	-	+2	us
M Delay Time	t_{DM}	-2	-	+2	us
CL 'LOW' Level Width	t_{WL}	35	-	-	us
CL 'HIGH' Level Width	t_{WH}	35	-	-	us

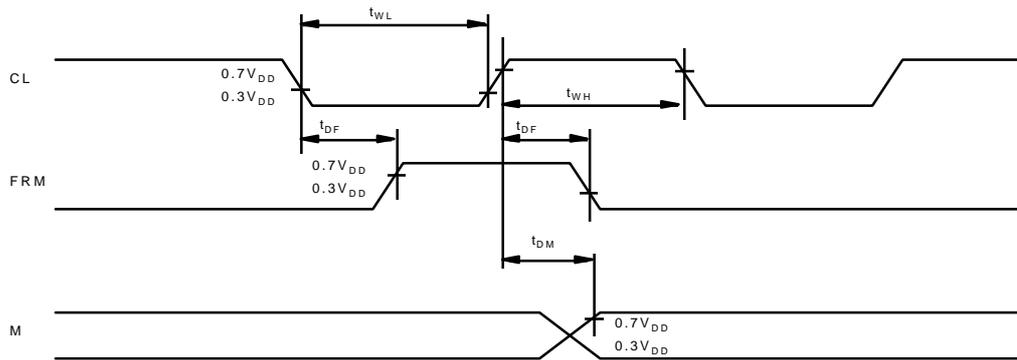


Fig 2. Display control signal waveform