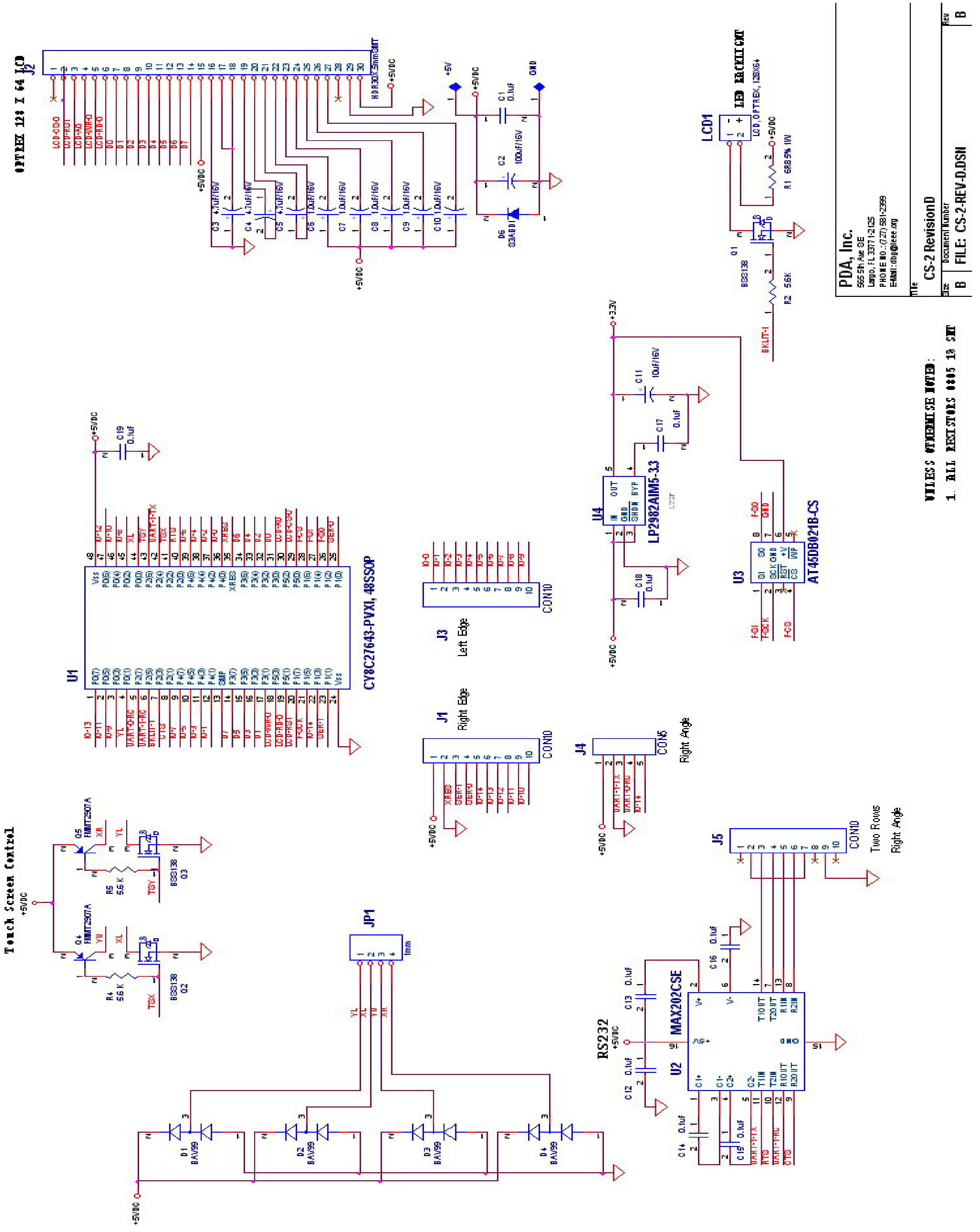
CS-2 Table of Contents

CS-2B Schematic



Touch Screen Display Module Startup Instructions

Initial Testing:

1. Remove module from packaging.

2.Connect +5 volts and ground to the module.
   A. Through connector J4 pin 1 to +5 volts and pin 2 to ground.
   B. Through connector J1 pin 1 to +5 volts and pin 2 to ground.

3. Turn on power.

4. Press anywhere on the screen to go into the local control mode.

5. After finishing a tour of the local control mode, you should be back to seeing the logo on the screen.

6. You can now use the RS232 level serial interface to test the display commands.

Testing the serial interface:

1. If you want to connect to a PC, attach the cable with DB9 connector to your PC and the other end to the ten pin black connector in the left bottom side of the controller board.

2. Use a terminal program such as terminal.exe or hypertext to communicate to the display module. You may also download a utility program, CS2UTILexe, from www.simmetry.com that provides a simple terminal program along with the ability to download screens and programs to the module.

3. The correct serial parameters are no parity and one stop bit. The baud rate is 9,600, 19,200 or 38,400. Send a "*" and carriage return at each baud rate and wait for a ">" to be returned. You are now at the current baud rate, which may be changed with a software command.

4. Please go to www.simmetry.com for instructions on the available Commands.

5. To use an RS-232 connection at TTL levels, such as from a Basic Stamp, PIC or 8051, connect the RS-232 output of your microprocessor to pin 4 of J4 and the RS-232 input to pin 3 of J4. Connect a wire to pin 5 of J4 if you wish to use a signal level rather than a serial Character to indicate when a command has finished executing. Connector J4 is on the middle of the bottom of the board. J4 may also be used for power connection.

CS-2 Display Module Commands

Go to www.simmetry.com for latest information with examples and pictures.
Type in the first character
followed by any other parameters and then a carriage
return to start the command. The display returns a
">" when ready for the next command.

; - Automatic touch alert
A > is sent when the controller finishes executing a command. The signal level on pin 5 of J4 is also pulled to a logic 1 after the display finishes executing a command. If a ";" is sent after a menu is created, then the > and pin 5 level shift will be held until a button is pressed. If you have an interrupt driven processor, the ";" command will allow you to ignore the display until a user presses the screen and asserts your interrupt through your UART or through pin 5 of J4..

“;” - Display holds off sending the usual > and pin 5 response until the touch screen is pressed. A “K” would then be sent to receive the touch screen character.

= - Baud rate
Send a “=” followed by a digit from 0 through 2 to set the baud rate of the RS-232 communication port. 0 = 9600 baud, 1 = 19,200 baud and 2 = 38,400. The baud rate is saved and will be the baud rate on power up or after a reset. To determine the baud rate of a display, send a “*” at each baud rate and wait for a “>” to be returned.

“=2” - Sets the display to 38,400 baud.

: - Run calibrate process - applies only to touch screen
Send a ":" followed by a carriage return to start the calibration process. A screen will be displayed asking that a dot in the lower left corner be pressed. After that is done, a screen will be displayed asking that a dot in the upper right corner be pressed. After that is done, the screen is calibrated and the values are permanently stored in the flash memory. A “#” may be sent after the “:” to return to default settings.

“:” - Execute the calibration function.

“:#” - Restore default touch screen settings.

* - Test response
Send a "*" followed by a carriage return to have the controller respond with a ">". This command is used to check if the controller is finished or locked up. Also useful to check if you are using the correct baud rate. Set one baud rate, send "*" and timeout if there is no response. Then try another supported baud rate. This method is used in the utilities program.

“*” - Returns a > if the display is active and set at the same baud rate as you processor.

[- Display stored BMP screen

Display a previously stored screen. Send a “[“ followed by a ″A″ through “Z″ character. The “A″ screen will be shown on power up or reset. The screen is 128 by 64 pixels and may be downloaded from a BMP type file using the utilities program.

“[R″ - Displays the screen located at record R in the flash memory.

] - Save current screen as BMP screen

Save the currently displayed screen. Send a “]″ followed by a “A″ through “Z″ character to save the screen for later retrieval. Use "[" to display it.

“]L″ - Saves currently displayed screen to location L in the flash memory.

(- Set normal display mode

Sets display to normal mode with white letters on blue background. Use ")" to set to reverse display mode.

“(“ - Sets the display to normal mode and saves that mode as the default on start up.

) - Set reverse display mode

Sets display to reverse mode with blue letters on white background. Use "(" to set to normal display mode.

“)“ - Sets the display to reverse mode and saves that mode as the default on start up.

@ - Display touch screen location

Displays x and y location of touch screen contact. Sending a "@" toggles the location display on and then off.

“@″ - Displays the location of any point pressed on the screen.

^ - Display revision screen

Displays the current revision number.

“^″ - Displays the revision and copyright statement for the display’s software.

A - Run a built in touch screen function

Selects the number entry function. The "AN" command causes the controller to display the number entry menu. The controller does all the work, allowing the user to enter numbers, a decimal point, change to plus or minus, backspace to remove characters and to press the E for enter key. Your program uses the "K" command to repeatedly check the latest button pressed. When a E is returned, a "KS" command is sent to return the complete numeric string that the user entered.

Repeatedly using the "K" command will return the appropriate character if a button is pressed or a # if no button is pressed.
The “AP” command does the same as the “AN” command except that entered numbers are shown as a “*”. This allows a password to be entered.

“AN” - Display and activate the number entry function.

“AP” - Display and activate the password entry function.

B - Selects one of the touch screen templates

See www.simmetry.com/template.asp for an extensive discussion of templates. Template A allows selecting from up to 4 lines, template C allows changing the value of a parameter and template D allows up to 16 buttons. Send a "B" followed by the template character to activate a particular template. Follow the template letter with a "#" and the template will be activated without writing on the screen. This allows you to make your own touch screens. Clearing the screen, turns off any active template.

“BA” - Selects and activates template “A" and writes that template over the current screen.

“BA#” - Selects and activates template “A" without writing that template over the current screen.

“BA#M” - Selects and activates template “A" after loading screen “M”. Does not write template “A” over screen “M”.

C - Clears the screen to its background color.

“C” - Clears screen to background color.

D - Read switch data
A “D” reads or writes digital data to eight pins in connector J3. The pins use normal TTL levels and are pulled up to +5 volts through a 5.6 k ohm resistor. All pins are initially set to high level so that they may be used as input pins. Pulling a pulled up pin to a low level will not damage it, but trying to drive an input pin to a high level when it is set to a low level may damage it. All of the eight pins may be used as input or output, but any intended as input should be set to a high level using the “DO” command.
“D” returns the hex value of pins IO-0 through IO-7 on connector J3, the 10 pin header on the left side of the controller board. A “DP” will output the following hex number to the eight pins. A “DM” returns either a “#” if no key in a four by four switch matrix or a character from “A” through “P” starting at IO-0 connected to IO-4 and ending with IO-3 connected to IO-7.

“D” - Returns “3E” with pins IO-7, IO-6 and IO-0 low.”

“DM” - Returns a “B” if the second switch in the top row is closed.

“DPF0” - Sets pins IO-7 through IO-4 high and IO-3 through IO-0 low.”

E – Echo control
Send a “EX” and the display will stop echoing input characters and will not send error messages. Send a “E” and echoing will be restored. Reset or power up has echoing on by default.

“E” - Turn on echo.

“EX” - Turn off echo.

F - Set display contrast value
A "F" followed by a value from 20 to 50 sets the contrast value for the display. This value is set to 32 in final test, but slight changes may improve the resolution of the display.

“F30” - Set contrast to a value of 30 and save in EEPROM.

G - Control the output of a sequence of data points through an eight bit D to A converter. The sequence of data points is download from a PC using the download utility. The sequence may have up to 63 points and the values for each point must be between 0 and 255. The first value of the download sequence is the total number of data values in the sequence.
Data points are output every every 50 milliseconds. The outputted data will be 0 volts for a zero data value and 5 volts for a data value of 255.

“G” - Start sequence from pin 9 of connector J1.

“GR” - Start sequence from pin 9 of connector J1 and keep repeating sequence until receiving a “GX” command or a reset command.

“GX” - Stops a sequence.

H - Set the language offset. A project requires six screens for a particular application. When the application is finished, the set of six screens is translated into two other languages. The three sets of screens are saved in the external flash on the controller board in sequence. When the application is started, the user is requested to select a language. If the user selects the original language, a “H0” is sent to set the language offset to zero. If the second language is selected, a “H6” is sent to select the second set of screens. If the third language is selected, a “H12” is sent to select the third set of screens. When the application program requests a screen, the offset is automatically added to the screen character to display a screen from the appropriate set.
The utility program may be used to download all of the screens during the display module’s initialization.

“H6” - Sets the screen offset value to be 6 screens passed the screen requested.

I - Set display intensity
A "I" followed by a value from 0 to 100 sets the intensity of the backlight.

“I80” - Set backlight to 80% of maximum value.

J - Load eight bit digital to analog converter.

“J128” - Loads the DAC with 128 giving an output on pin 10 of J3 of 2.5 volts.

K - Key command

A "K" returns a # if the screen has not been pressed and the appropriate character if a template is active. The "K" command also sets the saved character to a # so that the next "K" command will return a # unless a new button has been pressed. A "KS" returns the stored data string created by the number entry function.

“K” - Returns last pressed location on screen or # if no area has been pressed since the last time the K command was sent.

“KS” - Returns any characters stored in the string array.

L - Draw line, box or filled box

This controller simplifies line drawing by using a reference system that sets the 0,0 point at the lower left corner of the area. A “L” followed by a “#” causes the point, line, box or filled box to be drawn using the background color. It may be used to erase a previously drawn line or a filled box may be used to clear portions of the screen.

"L10,12" - displays a single point at x=10 and y=12.

"L10,12,20,25" - displays a line from x=10 and y=12 to x=20 and y=25.

"L-45,20" - displays a line from the last plotted point to x=45 and y=20.

“LB10,12,20,25” – displays a box with corners at x=10 and y=12, x=10 and y=25, x=20 and y=12 and x=20 and y=25.

“LBF10,12,20,25” – displays a filled box with corners at x=10 and y=12, x=10 and y=25, x=20 and y=12 and x=20 and y=25.

“L#BF10,12,20,25” – displays a filled box, in background color, with corners at x=10 and y=12, x=10 and y=25, x=20 and y=12 and x=20 and y=25.

N - Write a number command.
A "N" command is used to write a number string to the display. The "N" followed by s for small, m for medium, l for large or h for huge sets the font. The next character sets the row. Next one or two characters sets the column and is always followed by a comma. The rest of the line until a carriage return is displayed at the selected row and column in the selected font. The advantage of this command over the "T" command is that this command automatically removes leading zeros. When you use "DEC3",

Basic Stamp adds leading zeros if needed to give 3 digits. The "N" command inserts blanks for any leading zeroes.

“NM11,028” - Displays “28” starting at location row 1 column 2.

O – Draw a circle
Send an “O” followed the circles center x and y and then the radius of the circle.

“O63,32,10” – displays a circle centered at x=63 and y=32 with a radius of 10.

P - Automatically plot points

The full screen autoplot uses the lower half of the screen for the plot data. The height is a maximum of 32 pixels and it's length is 128 pixels. The plot feature has a continuous mode as well as a single point mode. Each mode may display a single analog point or four digital points. The digital points represent the one or zero state of an external pin while the analog plot represents the voltage value on an external pin. The upper half of the screen may be changed at any time without effecting the lower half of the screen.
To plot in the single point analog mode, clear the screen and write any text to the upper half of the screen. Plot each data point by sending a "PA" followed by a number with a value between 0 and 31. The controller moves to the next pixel column and connects the last point to the new point with a line and clears the next three columns of pixels. This clearing moves ahead of the plotted data and shows where the new data ends. When the data reaches the end of the screen, it starts over at the beginning. Autoplotting is an easy way to simulate
a strip chart recorder.
To plot in the single point digital mode, clear the screen and write any text to the upper half of the screen. Plot each set of points by sending “PD” followed by a single hex character, “0” through “F”. The four points are plotted as lines from the previous set of points. The controller moves to the next pixel column and connects the last set of points to the new set of points and clears the next three columns of pixels. This clearing moves ahead of the plotted data and shows where the new data ends. When the data reaches the end of the screen, it starts over at the beginning.
The continuous mode draws the same type of display as does the single mode, but, in the continuous mode, the data is automatically sampled every 50 milliseconds. This rate allows for continuous monitoring of action within a circuit.

“PC1” - Continuously plot data from pins 4, 5, 8 and 10 of connector J1.

“PC2” - Continuously plot data from pins 5 to 8 of connector J3.

“PC3” - Continuously plot data from pins 1 to 4 of connector J3.

“PC5” - Continuously plot data from pin 7 of connector J1.

“PC6” - Continuously plot data from pin 9 of connector J1.

“PCX” - Stops any continuous plot.

“PD9” - Plots a 1, 0, 1 and 0 set of values at the current location and automatically moves to the next location.

“PA25” - Plots a line from the previous value to the 26th pixel on the display.

R - Reset display command
A "R" followed by a carriage return resets the display module.

“R” - Resets the display module.

S - Retrieve the x, y coordinates from the touch screen
A "S" followed by a carriage return retrieves the x any y values of the point currently pressed on the touch screen. The values are "000,000" if no point is being pressed. If the touch screen is being pressed, the values might be 115,065 which is x = 115 and y = 65. The x and y directions are divided into 0 to 127 units in the x direction and 0 to 63 in the y direction.

“S” - Returns “000,000” if no point is currently being pressed on the touch screen or the actual location being pressed.

T - Write text string to display
A "T" command is used to write a string to the display. The "T" followed by s for small, m for medium, l for large or h for huge sets the font. The next character sets the row. Next one or two characters sets the column and is always followed by a comma. The rest of the line until a carriage return is displayed at the selected row and column in the selected font. The column width for small characters is 6 pixels while for all other characters is 8 pixels for flexibility in placing the larger fonts.

“TS15,Line one” - Displays “Line one” along the top row of the display starting at column 5”. There are 21 columns with the small font.

“TM23,Medium font” - Displays “Medium font” on rows 2 and 3 staring at column 3. There are 16 columns with the medium font.

“TL21,Large font.” - Displays “Large fo” on rows 5 to 8 starting at column 1. There are 8 columns with the large font.

“TL11,Stop” - Displays “Stop” on rows 1 to 8 starting at column 1. There are 4 columns with the huge font.

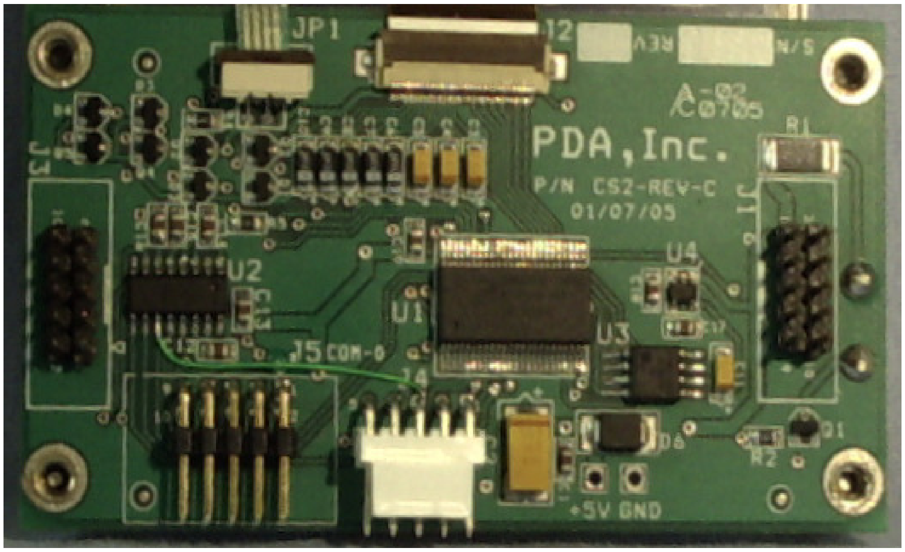
W - Set wait before responding delay
A "W" followed by a number between 0 and 250 sets a minimum delay in milliseconds before the controller responds to a command in order to give the Basic Stamp time to start receiving data.

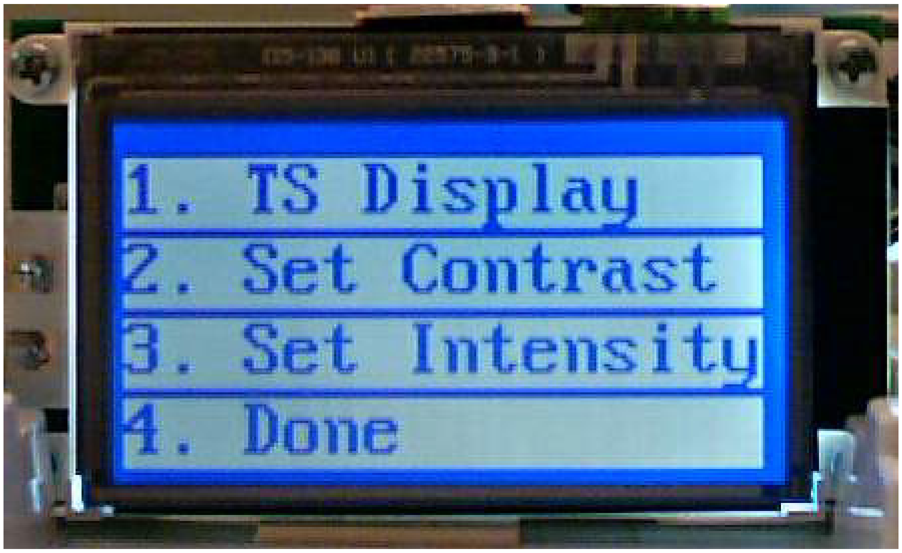
“W50” - Sets the delay to 50 milliseconds after completion of each command.

X - Read the voltage on pin 7 of J1, expressed as a value from 0 to 255 for 0 to +5 volts input.

“X” - Returns current value of voltage at pin 7 of J1.

CS-2 Display Module Pictures





```
' {$STAMP BS2}
'************************************************************************
'*  Basic Stamp Activity Board Display Demonstration Program           *
'*                                                                      *
'*  21 April 2005                                              (BS-2)   *
'*                                                                      *
'************************************************************************
  DIRS = %0001000000000000 ' Set direction for all bits
  x VAR Byte
  y VAR Byte
  z VAR Byte
'  Set pin 15 to rs-232 input
  SERIN 15,84,10,wait_start,[WAIT(">")]
'  Send carriage returns until CS-2B responds
wait_start:
  SEROUT 14,84,[CR]
  SERIN 15,84,1000,wait_start,[WAIT(">")]
'  Reset CS-2B
  SEROUT 14,84,["R",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
'  Give it time to reset
  DEBUG "Start",CR
  SEROUT 14,84,["^",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
  SEROUT 14,84,["ts11,123456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts21,223456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts31,323456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts41,423456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts51,523456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts61,623456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts71,723456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["ts81,823456789012345678901",CR] ' Twentyone columns
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["]D",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
  SEROUT 14,84,["[D",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
s1:
  TOGGLE 12
  PAUSE 1200
  GOTO s1
  END
wait_char:
'  SERIN 15,84,1000,wait_timeout,[WAIT(">")]
  PAUSE 3
  IF IN13 = 0 THEN wait_char
  RETURN
wait_timeout:
  DEBUG "Timed out"
  RETURN

wait_ds:
  SERIN 15,84,1000,wait_timeout,[WAIT("$")]
  RETURN

wait_long:
  SERIN 15,84,8000,wait_timeout,[WAIT(">")]
  RETURN

'**********************************************************************
' {$STAMP BS2}
'************************************************************************
'*  Basic Stamp Activity Board Display Demonstration Program            *
'*                                                                      *
'*  21 April 2005                                              (BS-2)   *
'*                                                                      *
'*  Figures example                                                     *
'*                                                                      *
'*  Line, box, filled box reversed filled box and circles               *
'*                                                                      *
'************************************************************************
  DIRS = %0001000000000000 ' Set direction for all bits
'  Set pin 15 to rs-232 input
  SERIN 15,84,10,wait_start,[WAIT(">")]
'  Send carriage returns until CS-2B responds
wait_start:
  SEROUT 14,84,[CR]
  SERIN 15,84,1000,wait_start,[WAIT(">")]
'  Reset CS-2B
  SEROUT 14,84,["R",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
'  Give it time to reset
  DEBUG "Start",CR
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 1000
  SEROUT 14,84,["lbf20,20,110,40",CR] ' Filled box
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["lb10,15,100,55",CR] ' Box
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["l0,63,120,10",CR] ' Line
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["O100,32,20",CR] ' Circle
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["O30,45,10",CR] ' Circle
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["l#bf25,25,90,35",CR] '
  GOSUB wait_char ' Wait for command response
  PAUSE 10000
  SEROUT 14,84,["R",CR] ' Reset display
  GOSUB wait_char ' Wait for command response
s1:
  TOGGLE 12
  PAUSE 1000
  GOTO s1
  END
wait_char:
  PAUSE 3
  IF IN13 = 0 THEN wait_char
  RETURN




'**********************************************************************
' {$STAMP BS2}
'************************************************************************
'*  Basic Stamp Activity Board Display Demonstration Program            *
'*                                                                      *
'*  21 April 2005                                              (BS-2)   *
'*                                                                      *
'*                                                                      *
'*
'*
'*
'*
'*
'*                                                                      *
'************************************************************************
  DIRS = %0001000000000000 ' Set direction for all bits
  x VAR Byte
  y VAR Byte
  z VAR Byte
  DEBUG "Start",CR
s1:
  TOGGLE 12
  PAUSE 100
  GOTO s1
  END

'**********************************************************************
```

```
' {$STAMP BS2}
'************************************************************************
'*  Basic Stamp Activity Board Display Demonstration Program            *
'*                                                                      *
'*  21 April 2005                                              (BS-2)   *
'*                                                                      *
'*                                                                      *
'*
'*
'*
'*
'*
'*                                                                      *
'************************************************************************
  DIRS = %0001000000000000 ' Set direction for all bits
  x VAR Byte
  y VAR Byte
  z VAR Byte
'  Set pin 15 to rs-232 input
  SERIN 15,84,10,wait_start,[WAIT(">")]
'  Send carriage returns until CS-2B responds
wait_start:
  SEROUT 14,84,[CR]
  SERIN 15,84,1000,wait_start,[WAIT(">")]
'  Reset CS-2B
  SEROUT 14,84,["R",CR]
  GOSUB wait_char ' Wait for command response
  PAUSE 2000
  SEROUT 14,84,["EX",CR]
  GOSUB wait_char ' Wait for command response
'  Give it time to reset
  DEBUG "Start",CR
' Menu 1
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm11,1. Menu 2",CR] ' Row 1
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm31,2. Menu 3",CR] ' Row 2
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm51,3. Menu 4",CR] ' Row 3
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm71,   Done",CR] ' Row 4
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["BA",CR] ' Set for 4 rows
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["]M",CR] ' Save screen as screen M
  GOSUB wait_char ' Wait for command response
  PAUSE 2000
' Menu 2
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm11,1. Blink once",CR] ' Row 1
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm31,2. Blink twice",CR] ' Row 2
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm51,3. Blink 3 times",CR] ' Row 3
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm71,   Done",CR] ' Row 4
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["BA",CR] ' Set for 4 rows
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["]N",CR] ' Save screen as screen N
  GOSUB wait_char ' Wait for command response
  PAUSE 2000
' Menu 3
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm11,1. Circle",CR] ' Row 1
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm31,2. Square",CR] ' Row 2
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm51,3. Circle & Square",CR] ' Row 3
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm71,   Done",CR] ' Row 4
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["BA",CR] ' Set for 4 rows
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["]O",CR] ' Save screen as screen O
  GOSUB wait_char ' Wait for command response
  PAUSE 2000
' Menu 4
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm11,1. 1 line",CR] ' Row 1
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm31,2. 2 lines",CR] ' Row 2
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm51,3. 3 lines",CR] ' Row 3
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm71,   Done",CR] ' Row 4
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["BA",CR] ' Set for 4 rows
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["]P",CR] ' Save screen as screen P
  GOSUB wait_char ' Wait for command response
  PAUSE 2000
' Menu 5
  SEROUT 14,84,["C",CR]
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm11,1. 1 filled box",CR] ' Row 1
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm31,2. 2 filled boxes",CR] ' Row 2
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm51,3. 3 flled boxes",CR] ' Row 3
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["tm71,   Done",CR] ' Row 4
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["BA",CR] ' Set for 4 rows
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["]Q",CR] ' Save screen as screen Q
  GOSUB wait_char ' Wait for command response
  PAUSE 2000
  SEROUT 14,84,["[A",CR]
  GOSUB wait_char ' Wait for command response
  SEROUT 14,84,["R",CR] ' Reset display
  GOSUB wait_char ' Wait for command response
s1:
  TOGGLE 12
  PAUSE 1200
  GOTO s1
  END
wait_char:
  SERIN 15,84,1000,wait_timeout,[WAIT(">")]
'  PAUSE 3
'  IF IN13 = 0 THEN wait_char
  RETURN
```

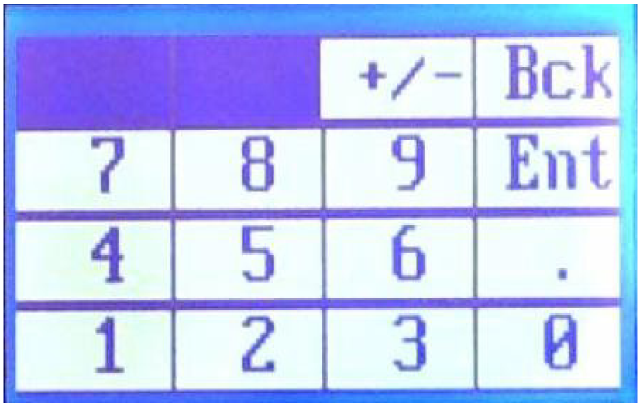
```
wait_timeout:
  DEBUG "Timed out"
  RETURN

wait_ds:
  SERIN 15,84,1000,wait_timeout,[WAIT("$")]
  RETURN

wait_long:
  SERIN 15,84,8000,wait_timeout,[WAIT(">")]
  RETURN

'***********************************************************************
```

CS-2 Template Function

Touch Screen Function

1. Number entry menu

Call this function and wait for your user to finish entering a number. Then retrieve the string with one command. The display does all the hard work and you save lots of coding! Control direction and speed with one finger.

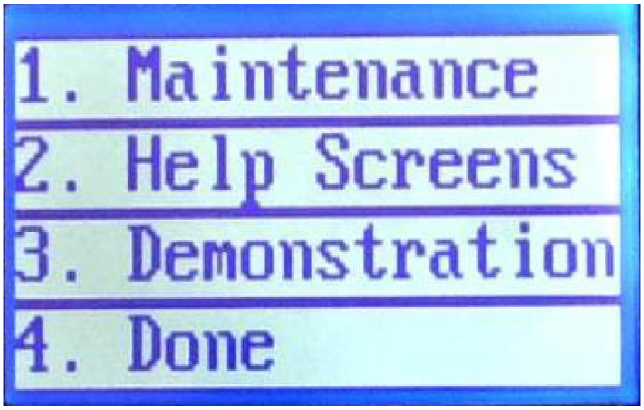


Touch Screen Templates

A touch screen menu is created by clearing the screen, writing the button labels, requesting a template, removing any unused buttons, adding any text you want and then waiting for a button to be pressed.

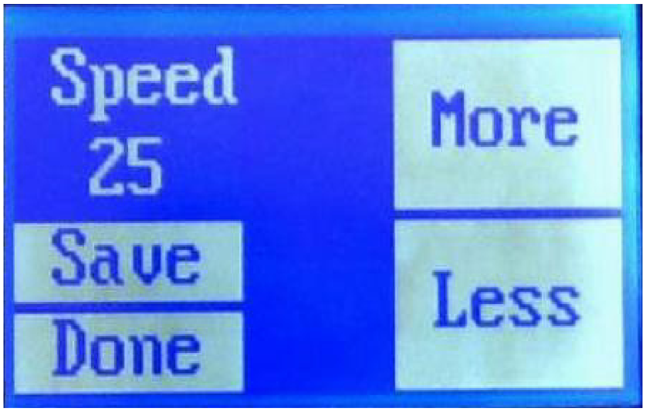
1. Select from a list template

Select from up to four lines. Remove lines you don't need.



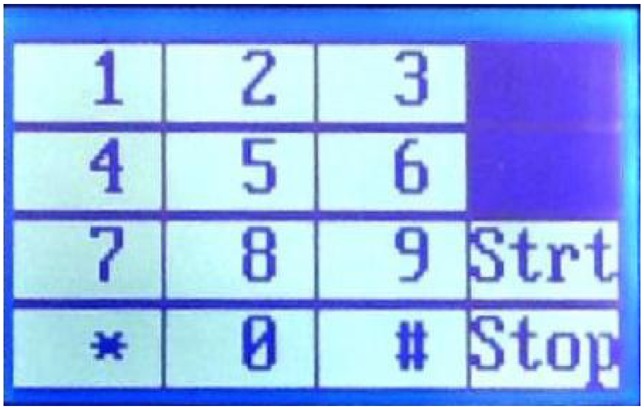
2. Change value template

Change the value of a parameter using this template.



3. Multiple button template

Create up to 16 buttons with your own labels.



Number entry menu
The AN command displays the number entry menu which acts like a calculator and enters the number on the screen while holding the number for later retrieval. You send a K command to retrieve the last button pressed. When a E is returned you know the user has pressed the enter key and now you can retrieve the number entered. Send a KS and the number string is returned.

The number entry menu handles the numbers, decimal point, backspace and changing between plus and minus. All you have to do is wait for the E button to be pressed and then retrieve the number.

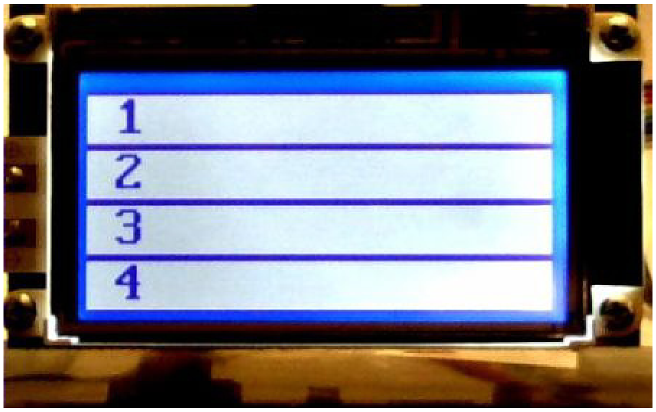
A password version of this control is available with the AP command. The display looks the same but only * will be displayed. Just like when entering a password in a Windows program. The KS command will return the actual characters entered by the user.

Touch Screen Templates
A touch screen template is highlighted areas of a screen that respond to being pressed. The display detects the pressed area and holds the character assigned to that area until you retrieve it. Sending a K returns the saved character and sets the saved character to a #. Receiving a # in response to a K command means that no button has been pressed since the last K command was received. The template always returns the same character for the same area on the screen no matter what text is written on the screen.

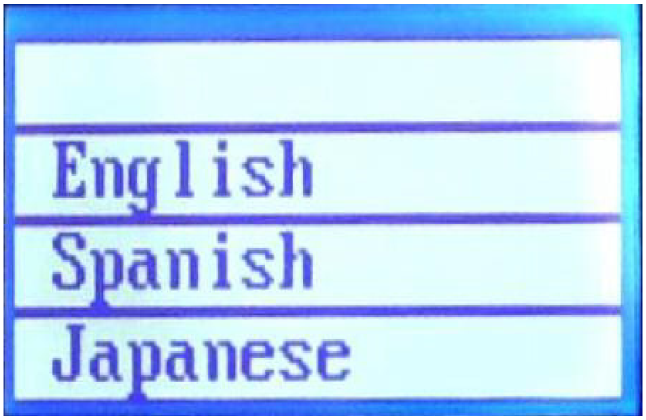
Note: A carriage return is assumed at the end of each line of code.

Template A - Select from a list

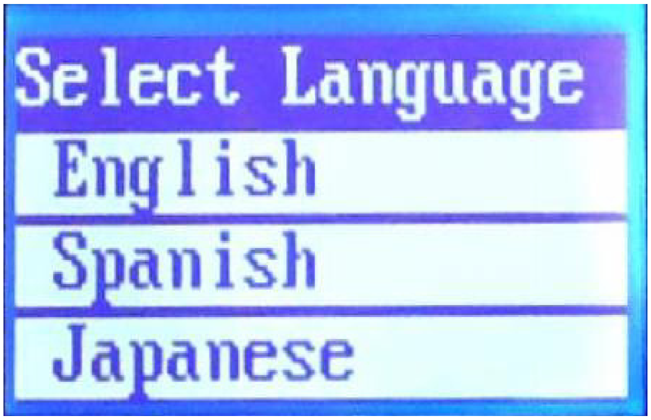
From 1 to 4 lines may be used as buttons. The buttons always return the number shown in the picture at left. Even if line one is not there, it still returns a 1 if pressed. If you ignore a button character, then it will seem to be inactive to you user. However, pressing a hidden button may be used for security purposes.



```
C ; Clear screen and write text
TM32,English
TM52,Spanish
TM72,Japanese
```

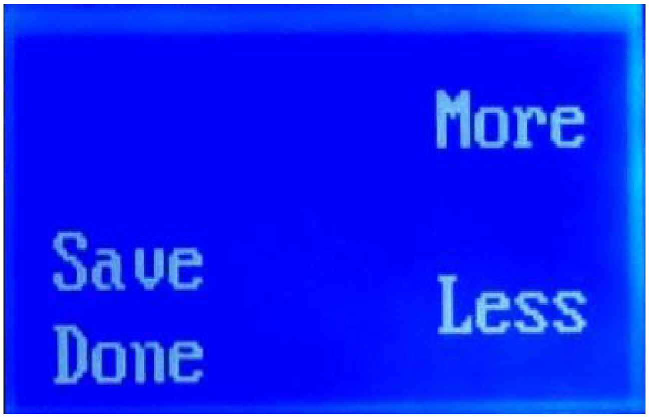


```
BA ; Overlay template A
; Remove first line as button by writing text
TM11,Select Language
```

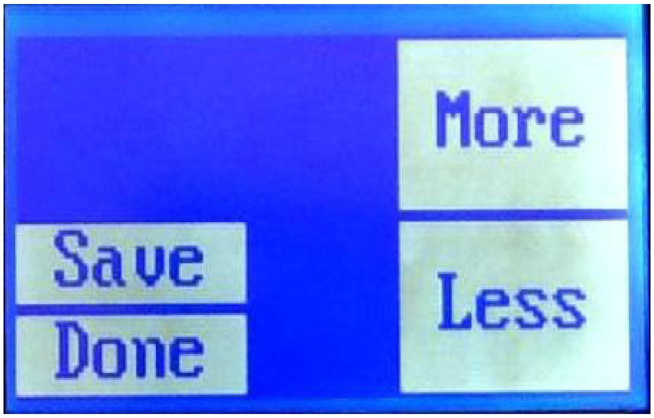


Template C - Change a value
Use the C template to increase or decrease the value of some parameter. This button does not require the user to release the button before a new value will be held. Otherwise, the user would have to press the menu many times for a large increase. The same value will be returned by the K command as long as the button is pressed.

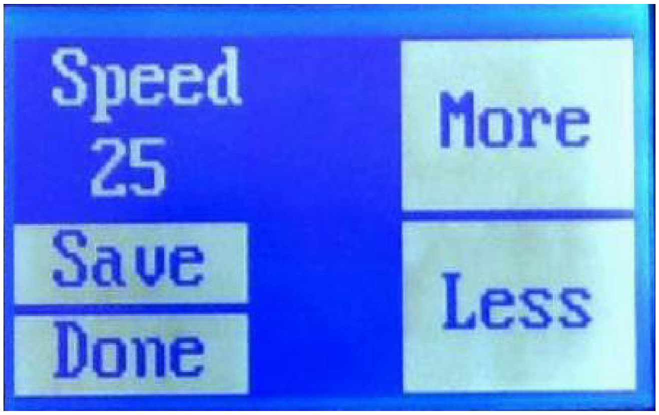
```
C ; Clear screen and write text
TM212,More
TM52,Save
TM612,Less
TM72,Done
```



```
BC ; Overlay template C
```



```
; Write text
TM12,Speed
TM33,25
BD ; Overlay template D
```

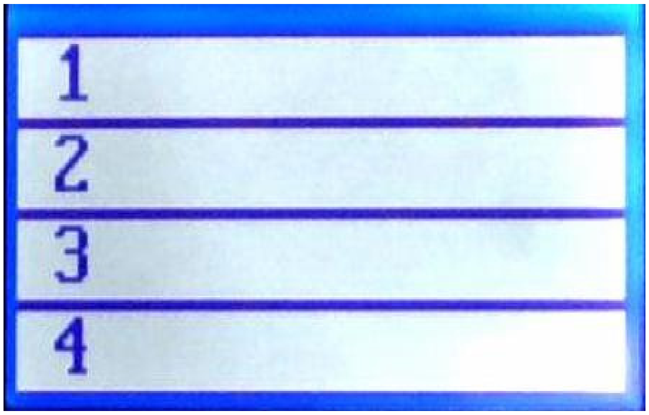


CS-2 Template Examples

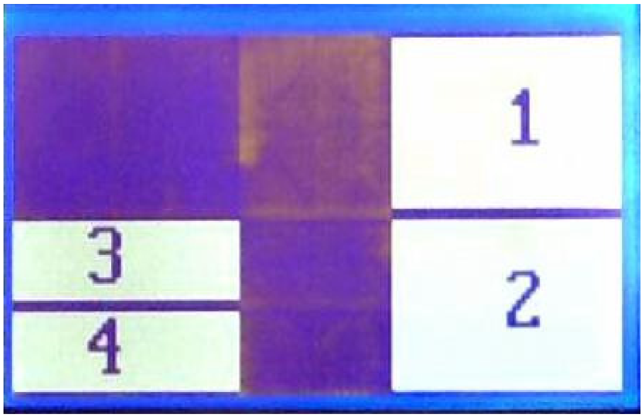
CS-2 List of Characters Returned from a Template

Each template shows the buttons created by that template and the character returned, using the "K" command, if that button is pressed.

Template A
Up to 4 buttons.



Template C
Change a value



Template D
Up to 16 buttons