

Application Note #5486

Interfacing a CrystalFontz Serial LCD Display with Galil

For applications that require a low cost user interface and do not need a lot of graphic capability, a serial LCD display is a good way to go. CrystalFontz makes a variety of LCD displays – but the CFA-634 unit is ideal for interfacing to a Galil controller RS-232 port. This unit accepts standard ASCII commands and displays them to the screen. This allows the programmer to output messages and display data from the controller right from the Galil program. A small sample library of Galil programs is shown at the end of this document that can be used to get up and running quickly. The following webpage provides full documentation on the serial LCD and should be referenced for complete information:

<http://www.crystalfontz.com/product/CFA634-NFA-KS.html>

A critical step after receiving the Serial LCD display is to place a small solder bridge across the jumper JPD as shown in the image below. This will disable the hardware handshaking so the controller does not have a problem with communicating to the LCD.

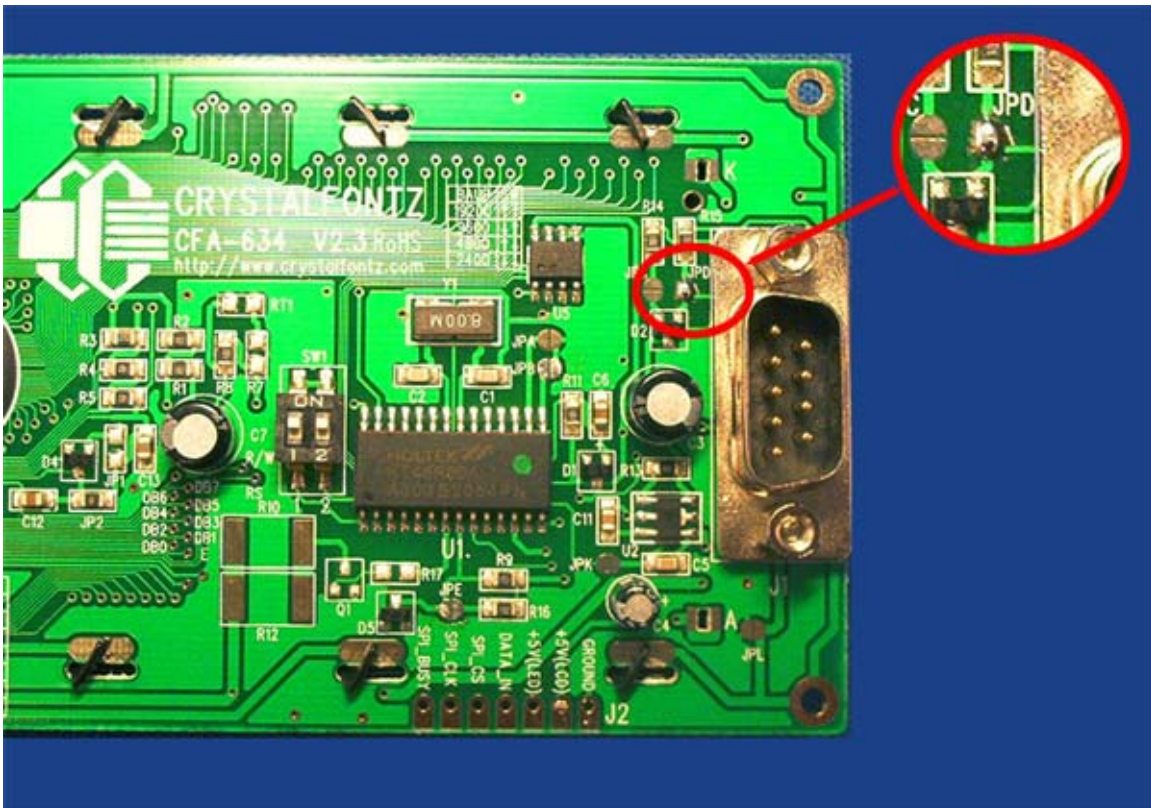


Figure 1: JPD jumper

The next step is optional. For some controllers, it is possible to order the unit with 5V power on Pin 9 of the RS-232 port. This 5V power is enough to power the LCD (Note: it is **not** enough power for the optional LED Backlight). If you have the -5Vpin9 option on the Galil controller, then the next step will allow you to use a single 9pin serial cable to power the LCD and communicate to it. Just solder a jumper wire from the back of pin 9 on the LCD RS-232 port and route it to the 5V(LCD) power input as shown in the image below.

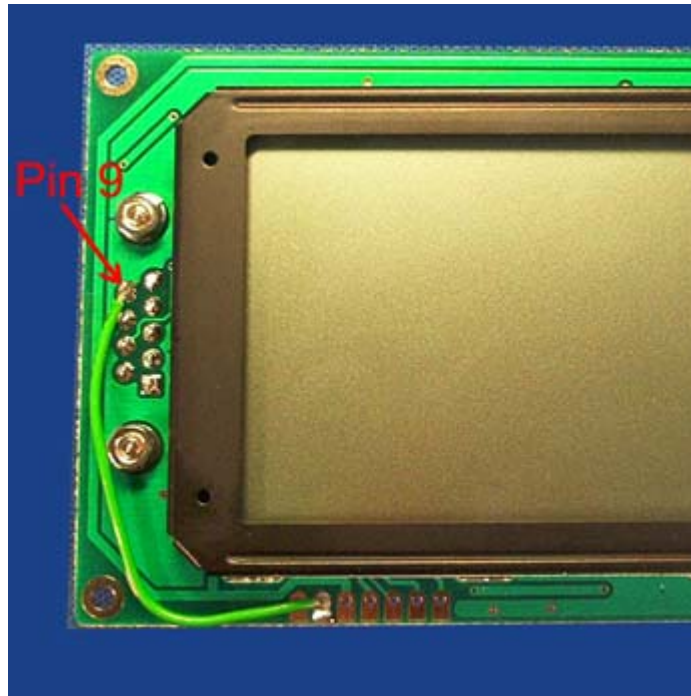


Figure 2: Pin 9 jumper wire

With the above modification, the serial cable will look like this for the RIO and Accelera:

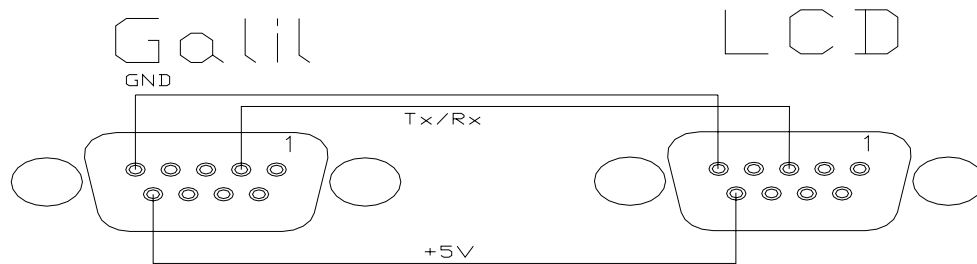


Figure 3: RIO Cable diagram when Galil ordered with +5V output on pin 9 (-5Vpin9)

If the above step is skipped, then an external power supply should be used to power the unit.

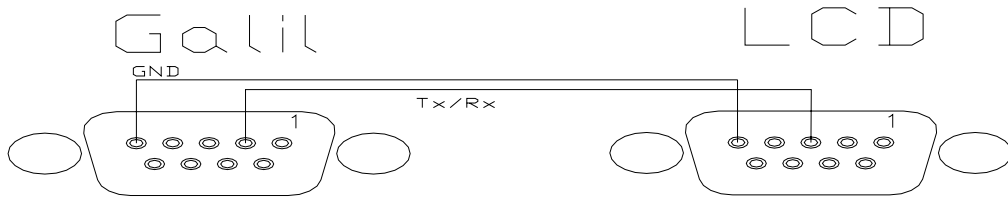


Figure 4: RIO Wiring diagram when external power supply connected via separate cable directly to LCD

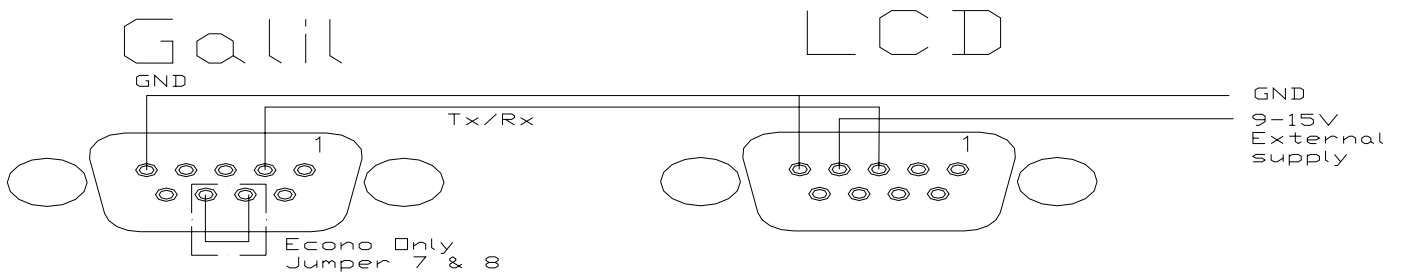


Figure 5: Wiring diagram when external power supply for LCD wired into serial cable

NOTE: for Econo series controllers such as the DMC-21x3, the hardware handshake lines - pins 7 and 8 (CTS and RTS) of the Galil side will need to be tied together.

Sample DMC Code Library for LCD Display

Display Analog Inputs

```
CFS;CW2
i=0;max=7
#screen1
JS#home
'display analog inputs
MG"AI0:" ,@AN[0]{F1.2} , " " {N}
MG"AI1:" ,@AN[1]{F1.2}
MG"AI2:" ,@AN[2]{F1.2} , " " {N}
MG"AI3:" ,@AN[3]{F1.2}
MG"AI4:" ,@AN[4]{F1.2} , " " {N}
MG"AI5:" ,@AN[5]{F1.2}
MG"AI6:" ,@AN[6]{F1.2} , " " {N}
MG"AI7:" ,@AN[7]{F1.2} {N}
WT200
JP#screen1
EN
#home
MG{^1}{N}; 'home cursor
EN
```

Display Digital Inputs

```
CFS;CW2
i=0;max=7
JS#blank
#screen1
JS#home
'display digital inputs
MG"Digital Inputs"
MG"B0:" {N}; 'bank 0
#1
MG@IN[i]{F1.0} {N}
i=i+1
JP#1,i<=max
WT200
i=0
JP#screen1
EN
#blank
MG{^12}{N}; 'BLANK SCREEN
EN
#home
MG{^1}{N}; 'home cursor
EN
```

Move Cursor

```
CFS;CW2
'set cursor position to 11
JS#home
WT1000
JS#showcsr
JS#movecsr
EN
#home
MG{^1}{N};'home cursor
EN
#movecsr
MG{^17},{^15},{^1}{N}
EN
#showcsr
MG{^6}{N}
EN
```

Display Big Characters

```
CFS;CW2
MG{^4}{N};'hide cursor
JS#blank
JS#bigchr
EN
v1=2
#bigchr
MG{^28},{^1},{^0},"3"{N}
MG{^28},{^1},{^5},"6"{N}
MG{^28},{^1},{^10},"2"{N}
MG{^28},{^1},{^15},"1"{N}
EN
#blank
MG{^12}{N};'BLANK SCREEN
EN
```

Display Real Time Clock with events (using RIO-471x2 Rev 1.0d firmware)

```
REM Must have RIO-471x2 with Firmware Rev 1.0d
REM Uses CrystalFontz Serial LCD
REM Assumes time has been previously set with RT command
CFS;CW2
JS#blank
JS#events
AT0
#time
i=0
JS#home
MG_RT2{F2.0},": ",_RT1{F2.0}{N}
MG": ",_RT0{F2.0}
#loop
IF((hour[i]=_RT2)&`
(min[i]=_RT1)&(sec[i]=_RT0))
JSevent[i]
ENDIF
i=i+1
```

```

JP#loop,i<events
AT-1000
JP#time
EN
#home
MG{^1}{N};'home cursor
MG{^4}{N};'hide cursor
EN

#events
DA*[]
events=2
DM hour[events]
DM min[events]
DM sec[events]
DM event[events];'subroutine pointer
'6:30:00 AM, turn on Pump 1
hour[0]=6
min[0]=30
sec[0]=0
event[0]=#pmp1on
'6:40:00 AM, turn off Pump 1
hour[1]=6
min[1]=40
sec[1]=0
event[1]=#pmp1off
EN

#pmp1on
SB1; 'turns on pump1
JS#movecsr
MG"PUMP 1 ON"
EN
#pmp1off
CB1; 'turns off pump1
JS#movecsr
MG"PUMP 1 OFF"
EN

#movecsr
MG{^17},{^1},{^1}{N}
EN
#blank
MG{^12}{N};'blank screen
EN

```

Set and Display Time of Day using RIO (DEPRECATED: use RIO-471x2 Rev 1.0d firmware w/ built-in clock)

```
'Program to set and display time
'DI0 increments time
'DI1 jumps to next step
CFS
CW2
step=0;ampm="A";hour=11;min=59;
XQ#count,1
JS#timeset
EN
#MAIN
JS#time0
#LOOP
JS#blank
JS#timer
WT50
JP#LOOP
EN
#timdsp
JS#blank
JS#update
MG"TIME:",hour{F2.0},":",min{F2.0},`
ampm{S1}
WT1000
JP#timdsp
EN
#time0
T1=TIME
#timer
T2=TIME-T1;T2=T2/1000
MG "TIMER: ",T2{Z4.2}
EN
#blank
MG{^12}{N};'BLANK SCREEN
EN
#timeset
'main loop
JS#blank
JS#sethour,step=0
JS#setmin,step=1
JS#ampm,step=2
JS#done,step=3
JP#timdsp,step=4
JP#timeset
#sethour
JS#blank
MG"SET HOUR"
MG"TIME:",hour{F2.0},":",min{F2.0},`
ampm{S1}
IF@IN[0]=1
hour=hour+1
ENDIF
IF hour=13
hour=1
ENDIF
WT400
EN
#setmin
JS#blank
```

```

MG"SET MINUTE"
MG"TIME:",hour{F2.0},":",min{F2.0},`
ampm{S1}
IF@IN[0]=1
min=min+1
ENDIF
IF min=60
min=0
ENDIF
WT400
EN
#ampm
MG"SET AM OR PM"
MG"TIME:",hour{F2.0},":",min{F2.0},`
ampm{S1}
IF(@IN[0]=1)
IF(ampm="A")
ampm="P"
ELSE
ampm="A"
ENDIF
ENDIF
WT400
EN
#done
'set start time
T1=TIME
IF(ampm="P")
hr_init=hour+12
ENDIF
IF(ampm="A")&(hour=12)
hr_init=0
ENDIF
IF(ampm="A")&(hour<12)
hr_init=hour
ENDIF
min_init=min
Tstart=(min*60*1000)+`
(hr_init*60*60*1000)
step=step+1
EN
#update
'convert TIME and add start time
'add code to reset Tstart every n days
T_adj=TIME-T1
T2=T_adj+Tstart
'how many mins
s=T2/1000
m=s/60
h=m/60
d=h/24
'FIND HOURS & AM/PM
days=@INT[d]
hr_now=h-(days*24)
hr_now=@INT[hr_now]
IF(hr_now>0)&(hr_now<12)
ampm="A"
ENDIF
IF(hr_now=12)
ampm="P"
ENDIF
IF(hr_now>12)&(hr_now<=24)
hr_now=hr_now-12

```

```
ampm="P"  
ENDIF  
IF(hr_now=0)  
hr_now=12  
ampm="A"  
ENDIF  
'END FINDING HOURS  
'FIND MINS  
hrs=@INT[h]  
min_now=m-(hrs*60)  
min_now=@INT[min_now]  
min=min_now  
hour=hr_now  
EN  
#count  
'code to detect next step  
#L1;WT100;JP#L1,@IN[1]=1  
step=step+1  
MG{EA}step  
WT1000  
JP#count
```