

**LCD INTEGRATION INTO BI-NODAL WIRELESS
TRANSCEIVER SOLUTION FOR INDUSTRIAL
MONITORING AND CONTROL**

by

Camille Lydia Leota Jaggernauth
B.A.Sc. Electronics (Honors), Simon Fraser University, 1998

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING SCIENCE

In the School
of
Engineering Science

© Camille Lydia Leota Jaggernauth 2008

SIMON FRASER UNIVERSITY

Spring 2008

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without permission of the author.

APPROVAL

Name: Camille Lydia Leota Jaggernauth
Degree: Master of Engineering Science
Title of Thesis: LCD INTEGRATION INTO BI-NODAL WIRELESS
TRANSCEIVER SOLUTION FOR INDUSTRIAL
MONITORING AND CONTROL.

Examining Committee:

Chair:

Dr. Ljiljana Trajkovic
Professor of Engineering Science

Dr. Steve Hardy
Senior Supervisor
Professor of Engineering Science

Todd Mawhinney
Supervisor
Senior Project Manager, OMNEX Controls Systems

Date Defended/Approved: April 04, 2008

Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <<http://ir.lib.sfu.ca/handle/1892/112>>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

ABSTRACT

The project report will discuss the integration of a LCD display module into a bi-nodal, wireless transceiver solution (transmitter and receiver pair) for industrial monitoring and control. The display module will be interfaced using the SPI protocol. GUI design considerations will be explored in the implementation of the man machine interface.

The project will implement monitoring and control functionality for analog, and digital, inputs and outputs. The wireless implementation details will be restricted to the application layer. At the network layer, the project will briefly discuss the proprietary, fully duplex, ORDEP protocol as well as the characteristic specifications of electronics in the mobile industry.

Keywords: Wireless Monitoring, Wireless Control, GUI Design considerations, LCD Integration.

Subject Terms:
LCD Integration, Wireless Control.

EXECUTIVE SUMMARY

In the Spring of 2008, OMNEX Controls unveiled a new transmitter, the T2600, shown below in Figure 1. The T2600 is a two way mobile radio remote control transmitter which features an interference free Frequency Hopping Spread Spectrum radio, operator keys (18 digital, 14 proportional), and a graphical LCD display used for industrial machine controls applications in concrete boom trucks, cranes, winches, vacuum trucks etc. The project work detailed the integration of the graphical LCD display into the transmitter design as well as the development of an initial concrete pumping application using the T2600 as well as R2160 receiver and D180 expansion modules (Figure 2).

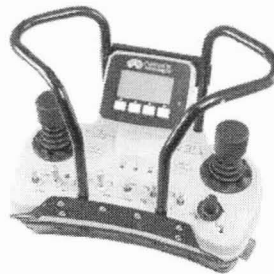


Figure 1 T2600 Transmitter.

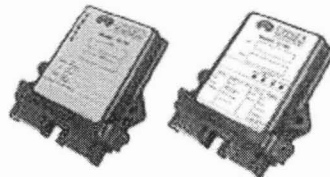


Figure 2 R2160 Receiver & D180 Expansion Module.

Mom & Dad,

Grandma & Grandpa,

Phebe,

Jason.

ACKNOWLEDGEMENTS

I would like to thank the following people for their support and encouragement during my graduate studies and especially this project.

I am excessively blessed in my personal support system, my faith, my family and my friends.

I would like to thank Dr. Steve Hardy for agreeing to be my senior supervisor – I am very appreciative of your kindness and flexibility.

I would also like to thank my industrial supervisor, Mr. Todd Mawhinney. I truly enjoyed working under your leadership on the challenging and aggressively scheduled T2600 project.

I would also like to thank my co-workers at OMNEX Controls Systems who have contributed to further this particular journey along.

I would also like to thank Dr. Jon Driver and the Office of the Dean of Graduate Studies, including Mr. Barry Williamson, for their understanding and decency, during some of the many challenges I faced while I pursued my graduate studies.

TABLE OF CONTENTS

| | |
|---|-----------|
| Approval | ii |
| Abstract | iii |
| Executive Summary | iv |
| Dedication | v |
| Acknowledgements | vi |
| Table of Contents | vii |
| List of Figures | ix |
| List of Tables | x |
| Glossary | xi |
| Chapter 1: LCD INTEGRATION | 1 |
| 1.1 Introduction | 1 |
| 1.2 Architecture | 2 |
| 1.3 LCD Dimensions | 4 |
| 1.4 T2300 GUI Interface Specifications | 5 |
| 1.4.1 Data | 5 |
| 1.4.2 Lists | 6 |
| 1.4.3 Screen | 6 |
| 1.4.4 Buttons | 7 |
| 1.4.5 Text | 8 |
| 1.4.6 Bitmap | 9 |
| 1.4.7 Menu | 9 |
| 1.4.8 Graphics | 10 |
| 1.4.9 Widgets | 10 |
| 1.5 ORDEP Wireless Protocol | 11 |
| 1.5.1 Frequency Hopping | 13 |
| 1.6 Software Layout | 15 |
| 1.6.1 Module Layout | 15 |
| 1.7 CHALLENGES | 18 |
| 1.8 TESTING | 19 |
| Chapter 2: APPLICATION DEVELOPMENT | 21 |
| 2.1 GUI Design Considerations | 21 |
| 2.2 Beta Application Coding | 22 |
| 2.2.1 Display Information | 22 |
| 2.2.2 Beta Application | 23 |

| | |
|---|-----------|
| 1.4.1 Power Up Screen..... | 26 |
| 1.4.2 Application Screen..... | 27 |
| 1.4.3 Diagnostic Screen..... | 28 |
| 1.4.4 Calibration Screen..... | 31 |
| 1.4.5 System Screen..... | 31 |
| Chapter 3: CONCLUSION..... | 34 |
| Appendices..... | 35 |
| Appendix A: ui_cfg.d Data Types..... | 35 |
| Appendix B: ui_cfg.d Enum Data Types..... | 36 |
| Appendix C: ui_cfg.d Bit Image..... | 37 |
| Appendix D: ui_cfg.d Bit Text..... | 38 |
| Appendix E: ui_cfg.d Screen Definition..... | 39 |
| Reference List..... | 40 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1 T2600 Transmitter. | iv |
| Figure 2 R2160 Receiver & D180 Expansion Module. | iv |
| Figure 3 Mechanical Design of T2600 Transmitter. | 1 |
| Figure 4 T2300 Transmitter. | 2 |
| Figure 5 T2600 Software Architecture. | 3 |
| Figure 6 LCD and LCD Carrier Board. | 4 |
| Figure 7 Common Concrete Pumping Functionality. | 23 |
| Figure 8 Wiring Diagram for R2160 Receiver Module. | 24 |
| Figure 9 Wiring Diagram for D180 Expansion Module. | 25 |
| Figure 10 Introduction Splash Screen. | 26 |
| Figure 11 Main Application Screen. | 27 |
| Figure 12 Application Screen on T2600. | 27 |
| Figure 13 Digital Input Screen. | 28 |
| Figure 14 Digital Input Screen on T2600. | 28 |
| Figure 15 Analog Input Screen. | 29 |
| Figure 16 Analog Input Screen on T2600. | 29 |
| Figure 17 Serial / Analog Input Screen. | 30 |
| Figure 18 Serial / Analog Input Screen. | 30 |
| Figure 19 Calibration Screen. | 31 |
| Figure 20 Password Screen. | 32 |
| Figure 21 System Settings Screen. | 32 |
| Figure 22 Factory Password Screen. | 33 |

LIST OF TABLES

| | |
|--------------------------------|----|
| Table 1 Data Types. | 5 |
| Table 2 List Types. | 6 |
| Table 3 Button Types. | 7 |
| Table 4 Font Types..... | 8 |
| Table 5 Text Types..... | 8 |
| Table 6 Bitmap Types..... | 9 |
| Table 7 Menu Types..... | 9 |
| Table 8 Drawing Elements. | 10 |
| Table 9 Widget Types..... | 10 |

GLOSSARY

| | |
|----------------------------|--|
| Base Control Unit | ORIGA receiver device physically attached to the machinery. It receives remote control data from the Remote Unit and performs the control functions of target machine. The bi-directional base units also transmit display and diagnostic data to the remote unit. |
| CAD | Computer Assisted Design |
| GUI | Graphical User Interface |
| MMI | Man Machine Interface |
| OMuD | ORIGA Multiple Data Set Protocol - the existing radio link communication protocol used by ORIGA devices. It is one-way, single byte data payload serial message data protocol. |
| ORDEP | ORIGA Remote Data Exchange Protocol . The new communication protocol is to replace the OMuD protocol. It is two-way, multiple byte data protocol. |
| ORIGA | OMNEX line of remote control devices, both remote and base units. |
| PCB | Printed Circuit Board |
| Remote Control Unit | ORIGA Transmitter device that the operator uses to manipulate the machinery. It sends operator commands to the base unit via a wireless link. The bi-directional remote units also receives display and diagnostic data from the base unit. |
| SDK | Software Development Kit |
| SPI | Serial Peripheral Interface developed by Motorola to provide a synchronous, low cost and simple interface between microcontrollers and peripheral chips. |

CHAPTER 1: LCD INTEGRATION

1.1 Introduction

In the Spring of 2008, OMNEX Controls unveiled a new transmitter, the T2600, shown below in Figure 3, primarily targeted at the concrete industry.

There are myriad sub-projects involved in the release of a new transmitter for e.g. enclosure concept development, CAD model design, mold making, hardware electrical design, PCB design, mechanical validation, prototype production, software development, automatic test equipment development. This project will focus specifically on the software development for the LCD integration of the T2600.

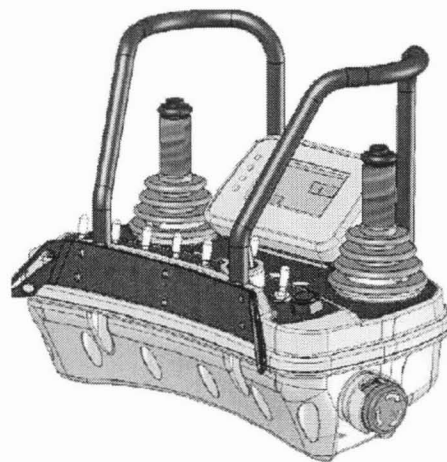


Figure 3 Mechanical Design of T2600 Transmitter.

1.2 Architecture

The proposed LCD integration architecture was based on porting the T2300 (shown in Figure 4) LCD display functionality. This solution allowed for the reuse of pre-existing components of the T2300 LCD design. The primary difference was that unlike the T2300, the display elements table structure was to reside on the transmitter and not the receiver.



Figure 4 T2300 Transmitter.

The final design solution required updates to the application as well the product and hardware software development library layers, shown below in Figure 5.

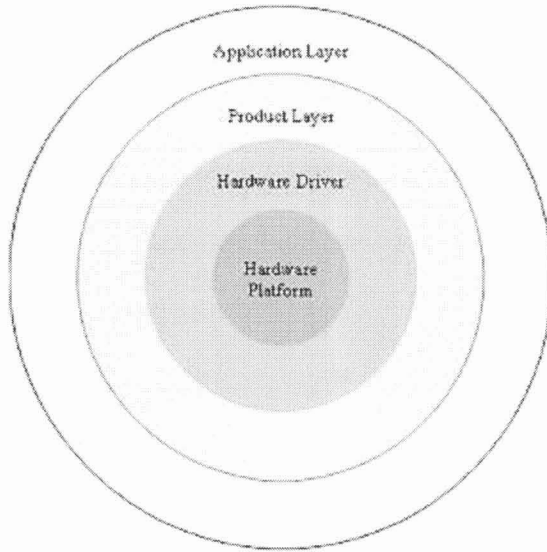


Figure 5 T2600 Software Architecture.

1.3 LCD Dimensions

The LCD used in this project was the **EA GE128-6N3V24** 128x64 Graphic Unit (shown in Figure 6) manufactured by Germany's Electronic Assembly. Key features of this display included:

- 3 Different font styles
- Zoom functionality on fonts
- Line, Dot, Bar Graphs, Bitmaps, predefined graphical elements.
- RS-232 input
- Programmable baud rates
- Fast Bus systems to reduced possible timing issues.
- 8 customizable digital I/Os [3]

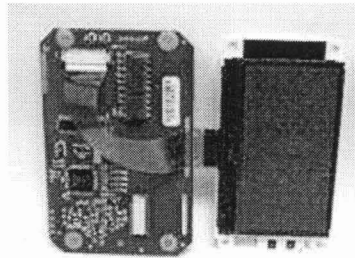


Figure 6 LCD and LCD Carrier Board.

1.4 T2300 GUI Interface Specifications

Because this was a pre-existing solution, a full suite of graphical library functions were already available from the previous T2300 development. The below sections gives a quick synopsis of some of the functionality that is currently supported.

1.4.1 Data

The data types shown below in Table 1 are supported by the graphic SDK. Appendix A and Appendix D also give a sample code listing [2] [5].

| Syntax | Type |
|-----------|----------------------------------|
| DT_BYTE | 8-bit unsigned |
| DT_SBYTE | 8-bit signed |
| DT_BIT_8 | 8-bit field |
| DT_WORD | 16-bit unsigned |
| DT_SWORD | 16-bit signed |
| DT_BIT_16 | 16-bit field |
| DT_LWORD | 32-bit unsigned |
| DT_SLWORD | 32-bit signed |
| DT_BIT_32 | 32-bit field |
| DT_CHAR | Character |
| DT_ENUM | Enumerated |
| DT_STRING | Zero-terminated character string |

Table 1 Data Types.

1.4.2 Lists

The list types shown below in Table 2 are supported by the graphic SDK. Appendix B and Appendix D also give a sample code listing [2] [5].

| Screen item that uses the list | List |
|--------------------------------|------------------|
| BIT_TEXT | Bit Text |
| ENUM_TEXT | Enumerated text |
| BIT_IMAGE | Bit image |
| ENUM_IMAGE | Enumerated image |

Table 2 List Types.

1.4.3 Screen

Each screen definition block starts with **BEGIN_SCREEN** and ends with **END_SCREEN** definition. Between **BEGIN_SCREEN** and **END_SCREEN** are definitions of graphical and text items. The syntax of the **BEGIN_SCREEN** and **END_SCREEN** is as follows below. Appendix E also gives a sample code listing [2] [5].

```
BEGIN_SCREEN(screen_name, parent_screen_name, prev_screen_name, next_screen_name)  
END_SCREEN(screen_name)
```

1.4.4 Buttons

The button types shown below in Table 3 are supported by the graphic SDK. Appendix E also gives a sample code listing [2] [5].

| Action | Description |
|----------------------------|--|
| UI_TBLS_ACT_NUL | Do nothing |
| UI_TBLS_ACT_NEXT_SCREEN: | Go to the next screen |
| UI_TBLS_ACT_PREV_SCREEN | Go to the previous screen |
| UI_TBLS_ACT_PARENT_SCREEN | Go to the parent screen |
| UI_TBLS_ACT_TOGGLE_BKLT | Toggle backlight |
| UI_TBLS_ACT_NEXT_MENU_ITEM | Go to the next menu item |
| UI_TBLS_ACT_PREV_MENU_ITEM | Go to the previous menu item |
| UI_TBLS_ACT_SEL_MENU_ITEM | Select the menu item |
| UI_TBLS_ACT_INC_VAL | Increase value |
| UI_TBLS_ACT_DEC_VAL | Decrease value |
| UI_TBLS_ACT_SET_VAL | Set value |
| UI_TBLS_ACT_EXIT_EDIT_MODE | Cancel editing, leave edit mode and go back to navigation mode |

Table 3 Button Types.

1.4.5 Text

The font and text types shown below in Table 4 and Table 5 are supported by the graphic SDK [2] [5].

| Font ID | Description |
|------------------|---|
| UI_TBLS_FID_4X6 | Small font (4x6 pixels – max 8 lines, 32 characters) |
| UI_TBLS_FID_6X8 | Medium font (6x8 pixels – max 6 lines, 21 characters) |
| UI_TBLS_FID_8X16 | Large font (8x16 pixels – max 4 lines, 16 characters) |

Table 4 Font Types.

| Text Type | Description |
|------------|--|
| VALUE_TEXT | Display numeric value to screen |
| BIT_TEXT | Display text dependant on binary state |
| ENUM_TEXT | Display text dependant on enum state |

Table 5 Text Types.

1.4.6 Bitmap

The image types shown below in Table 6 are supported by the graphic SDK [2] [5].

| Bitmap Type | Description |
|-------------|---|
| IMAGE | Display static image |
| BIT_IMAGE | Display image dependant on binary state |
| ENUM_IMAGE | Display image dependant on enum state |

Table 6 Bitmap Types.

1.4.7 Menu

The menu types shown below in Table 7 are supported by the graphic SDK [2] [5].

| FID | Description |
|-----------------|------------------------------|
| MENU_TEXT | Display scrollable menu item |
| MENU_VALUE_TEXT | Display editable menu item |

Table 7 Menu Types.

1.4.8 Graphics

The drawing elements shown below in Table 8 are supported by the graphic SDK [2] [5].

| Graphic | Description |
|---------|------------------------|
| POINT | Draw point |
| LINE | Draw line |
| RECT | Draw rectangle |
| RRECT | Draw rounded rectangle |
| CIRCLE | Draw circle |
| CIRCARC | Draw circular arc |
| POLYGON | Draw polygon |

Table 8 Drawing Elements.

1.4.9 Widgets

The custom controls shown below in Table 9 are supported by the graphic SDK [2] [5].

| Widget | Description |
|-------------|-------------------|
| THERMOMETER | Draw thermometer |
| BAR_RAMP | Draw bar ramp |
| BAR | Draw progress bar |

Table 9 Widget Types.

1.5 ORDEP Wireless Protocol

The wireless communication between the transmitter and receiver is point-to-point, where the receiver is always the initiator. There are 2 types of communication modes – configuration mode and control mode with their characteristic protocol and communication methods. The receiver and the transmitter will only work properly if they were validly configured [6].

The RF communication characteristics are listed below:

- baud rate: 4800
- start bit: 1
- bits per byte: 9
- stop bits: 1
- parity: none

Communication between the transmitter and the receiver can also be tethered (or wireline). The tethered communication characteristics are:

- baud rate: 38400
- start bit: 1
- bits per byte: 9
- stop bits: 1
- parity: none

The wireless message protocols use 9 bits per byte where the ninth bit is used to flag the first byte of the message frame – to allow the receiver to more easily detect the start of the message. Error checking uses a 16 bit CRC algorithm $X^{16}+X^{12}+X^5+1$ with seed = 0xFFFF [6].

In the configuration mode, the communication is half-duplex - the transmitter sends a request to the receiver unit and receiver replies back to the transmitter [6].

In control mode, the transmitter and the receiver exchange remote control, calibration, diagnostic, and machine status data.

Some of the data exchanged between the transmitter and the receiver are:

- ID numbers.
- Control data page number and the subscription bit-mask used by the transmitter to calculate the maximum message size and hop time for the RF communication.
- Transmitter uploads from the receiver the proportional output calibration data page number and the subscription bit-mask.
- Transmitter uploads from the receiver the base application data page <http://www.sfu.ca/dean-gradstudies/documents/EXAMMAST1.9.1-2-RevDec2006Fillable.pdf> number and the subscription bit-mask.
- Transmitter uploads from the receiver the base application data description.
- Transmitter downloads the hop ID to the receiver. The receiver calculates its hop table using the downloaded hop ID.
- Transmitter uploads the auto power off timeout from the receiver.
- Transmitter sends the end of communication message to the receiver [6].

1.5.1 Frequency Hopping

OMNEX radios are spread-spectrum frequency hopping radios. The Hop Link mechanism (frequency hopping over 63 radio channels after initial hop sequence synchronization between the transmitter and the receiver) is only employed in the control mode [6].

The complete message frame transmission and reception fit into the single hop time slot. The message size can vary, but is fixed during configuration phase to determine the minimum time transmitter and receiver have to sit at each frequency hop [6].

In control mode the communication is not half-duplex. The communication flow is controlled by the transmitter, the transmitter starts and then when it requires a response it sets the receive bit in the message data header and then switches to receive mode at the next hop. When the hop time has expired, it switches back to transmitter mode, regardless of whether data was received or not. The converse applies to the receiver – it waits to establish the Hop Link with the transmitter and then checks for the receive flag, if this is set, it transmits data at the next hop and then reverts to its natural mode which is receiving [6].

Loss of link is detected at both the transmitter and receiver if a certain number of messages in a pre-determined time period. To conserve battery life, the transmitter transmits at every nth hop when it is idle. The receiver detects link loss when a valid message has not been received for full hopping cycle (63 consecutive hops).

Transmitter transmissions are repeated for pre-defined number of times (typically 10 times), to increase chances of the information being received by the receiver [6].

1.6 Software Layout

The primary files associated with adding LCD support to the transmitter module are listed in the following sections.

1.6.1 Module Layout

1.6.1.1 ui_cfg.d

The ui_cfg.d file contains definitions of variables and graphical elements displayed on the screen. These variables are the graphical elements that were described in Section 1.4 above.

1.6.2 ui_tbl.c

The ui_tbl.c file is the initial file repository where the information defined by the developer in ui_cfg.d is first processed into a table format by a run-time tool. There are two separate tables defined and stored, the **main screen table** - (data types, button actions, strings, forms) and the **bitmap table** (bitmap pixel arrays, bitmap dimensions and screen locations). The screen table is stored locally in the transmitter's flash memory. A duplicate copy of the bitmap table is loaded into the LCD's memory for faster run-time loading and display. Because of memory limitations on the LCD's carrier board, the screen table is not duplicated like the bitmap table.

1.6.3 tbls_rec.c

The `tbls_rec.c` file is responsible for initializing and updating the graphical database record, specifically storing and retrieving information (screens, menus, button actions, strings, data values) from the graphical database record.

1.6.4 screen.c

The `screen.c` file is responsible for initializing the screen, drawing the screen, initializing and updating screen variables, configuring the LCD, switching screens by interfacing directly with the LCD hardware and the table database records defined in `tbls_rec.c`.

1.6.5 ui_man.c

The `ui_man.c` file defines the main GUI engine responsible for key handling and updating screen variables based on data received from the transmitter and receiver. It provides one layer of abstraction up from `screen.c`.

1.6.6 app_ui.c

The `app_ui.c` file details the button definitions, the GUI behaviour when idle, when the screen is redrawn, when a key is pressed, when a menu is selected, backlight and daylight sensor handling.

1.6.7 lcdapp.c

The lcdapp.c file is responsible for initializing the GUI, defining custom error strings and icons (outside of ui_cfg.d). ui_man.c, app_ui.c, lcdapp.c are the primary application interfaces.

1.7 CHALLENGES

The display used was the in-house OMNEX display. The firmware version is 1.3 D1SE. There were known limitations with this firmware version including:

1. If the maximum displayable characters per line is exceeded, the display will reboot.
2. Only a limited number of commands can be sent over the LCD SPI bus for processing or else the entire system will hang. The LCD data needs to be transferred over in chunks or blocks.
3. If a bitmap is too large to be displayed, then the bitmap will not be displayed.

1.8 TESTING

Testing involved designing a generic application and using some of the pre-defined default templates to demonstrate the functionality available. The generic application typified some of the functionality needed by existing concrete pumping customers and was geared towards a trade-shows demos. The following Section 2 describes the generic application that was developed for beta testing.

As per above, a subset of functionality was developed for two tradeshow, World of Concrete and ConExpo. This tradeshow application demonstrated the following key functionalities:

- Loading bitmaps – animation
- Designing custom controls – gauges
- Displaying text based alarms
- Displaying operational information including application specific information as well as more generic battery strength and signal strength.

Furthermore, there was the standard bench testing to ensure that all functionality worked as expected, including:

- Compensating for known device limitations, explored in Section 1.7 Challenges.

- Ensuring that there was no latency added to the overall operation of the device e.g., refreshing the displaying overloading the wireless data transmission.

CHAPTER 2: APPLICATION DEVELOPMENT

2.1 GUI Design Considerations

GUI design considerations that were incorporated into the MMI are:

- Adequate contextual information
- Storing of user and system parameters
- Brief, salient information
- User not overwhelmed by irrelevant data
- Data is easy to read, intuitively and usefully grouped, without too many distracting fonts
- Consistency in the design
- Layout always familiar to the user - keeping with expected information.
- Lists, tables
- Single, multi-variant graphic displays e.g., bar graph, strip chart display to present the information in a meaningful way
- Language and layout of the display as simple as possible to ensure readability and comprehension by all users.
- Icons should be simple, and concrete symbols should be designed to look like the object they represent and be distinguishable from others [7].

2.2 Beta Application Coding

2.2.1 Display Information

The information that is displayed for any concrete pumping application would depend on the customer and the application being written. This information can be either text based (output on or off, input on or off, value of proportional output) or icon based (e.g., flashing).

Based on previous applications and customer feedback, some of the information that is typically displayed include, Stroke Rate, RPM, Coolant, Oil Pressure, Fuel Tank Level, Totalizing Value, Hydraulic System Pressure, Hydraulic Temp, Hopper Gate Open, Forward/Reverse Pump State, Outrigger states, Pump Volume, Hydraulic System Oil Temperature, Pump On/Off, Boom functions, and Programming.

2.2.2 Beta Application

Figure 7 indicates the key functionality that is typically required for a transmitter concrete pumping application.



























| Symbol | Description | Symbol | Description |
|---|--|---|--|
|  | Boom 5th Section DOWN (on 5 section only) |  | Boom 5th Section UP (on 5 section only) |
|  | Boom 4th Section DOWN |  | Boom 4th Section UP |
|  | Boom 3rd Section DOWN |  | Boom 3rd Section UP |
|  | Boom 2nd Section DOWN |  | Boom 2nd Section UP |
|  | Boom 1st Section DOWN |  | Boom 1st Section UP |
|  | Boom Swing Counterclockwise |  | Boom Swing Clockwise |
|  | Horn |  | Outrigger Control |
|  | Engine Stop |  | Engine Start |
|  | Boom Speed (Slow) |  | Boom Speed (Fast) |
|  | Pump Volume (Slow) |  | Pump Volume (Fast) |
|  | RPM Increase |  | RPM Decrease |
|  | Pump Reverse |  | Pumping Forward |
|  | Boom Enable |  | Change Over |

Figure 7 Common Concrete Pumping Functionality.

Figure 8 indicates the key functionality - outputs that is typically required for a receiver concrete pumping application.

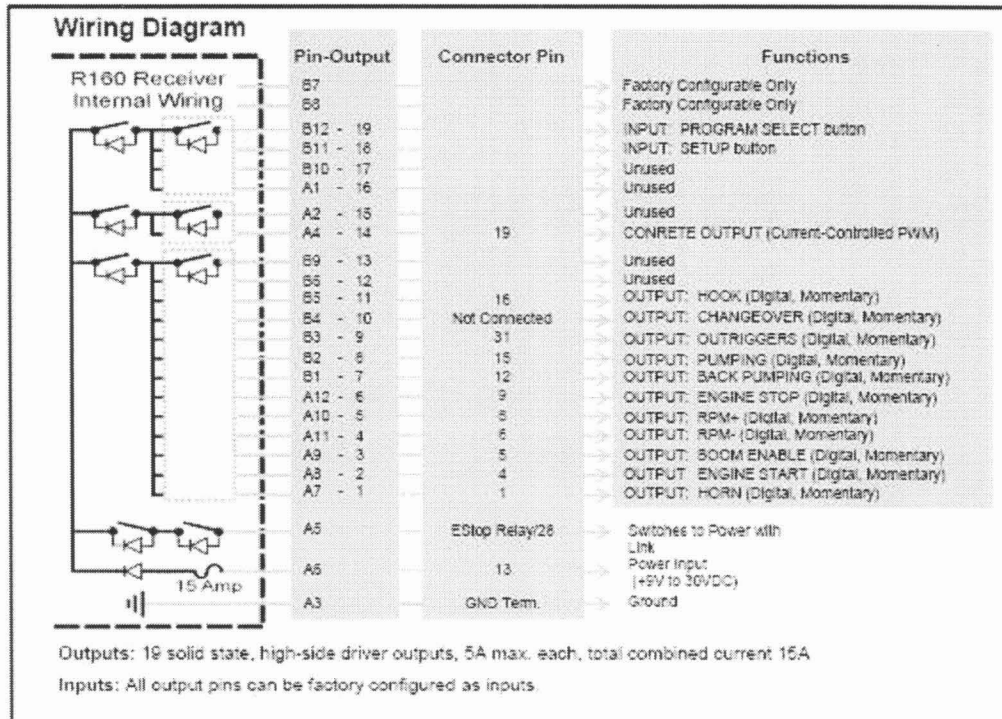


Figure 8 Wiring Diagram for R2160 Receiver Module.

Figure 9 indicates the key functionality – current control PWM (pulse width modulation) outputs that is typically required for an expansion module concrete pumping application [1].

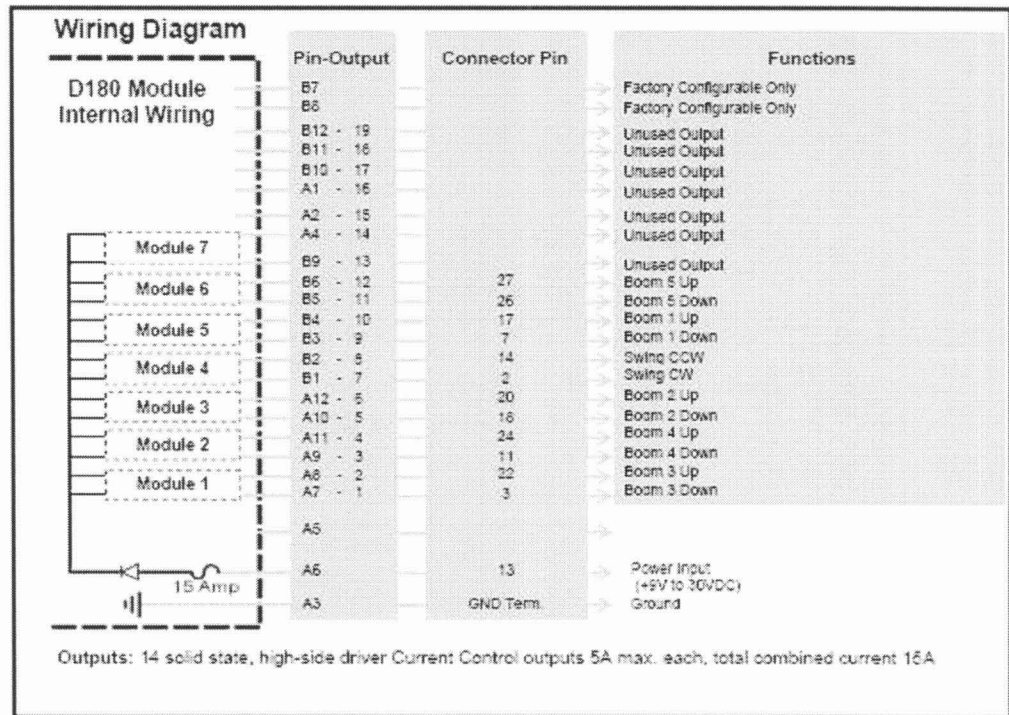


Figure 9 Wiring Diagram for D180 Expansion Module.

The following sections describe the beta application screens.

1.4.1 Power Up Screen

A “splash” screen with OMNEX logo appeared similar to Figure 10. (2 or 3 seconds) [4].



Figure 10 Introduction Splash Screen.

1.4.2 Application Screen

The application screen, shown in Figure 11 and Figure 12, showed application specific information e.g. pump on, RSSI (signal strength) as well as battery strength indicator [4].

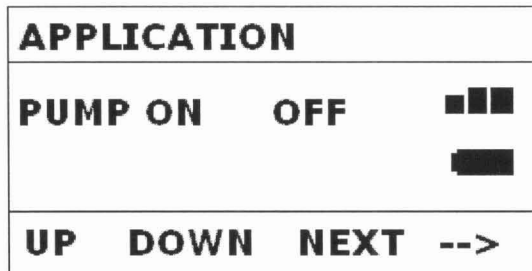


Figure 11 Main Application Screen.



Figure 12 Application Screen on T260.

1.4.3 Diagnostic Screen

The diagnostic screens followed the application screens. Screens were defined to show the values of the digital outputs/inputs, proportional outputs, proportional inputs [4].

Digital Input

The switches were mapped to the digital inputs. This screen, shown in Figure 13 and Figure 14, was developed to be used as a factory test to verify that all the switches were in fact working [4].

| DIGITAL INPUT | |
|----------------------|-----------------|
| INPUT [1-8] | 00000000 |
| INPUT [9-16] | 00000000 |
| INPUT [17-24] | 00000000 |
| INPUT [25-32] | 00000000 |
| NEXT --> | |

Figure 13 Digital Input Screen.



Figure 14 Digital Input Screen on T2600.

Analog Screen

The analog screen, shown in Figure 15 and Figure 16, show the raw analog input values.

| ANALOG INPUT | |
|---------------------|-------------------|
| INPUT 1: 0 | INPUT 5: 0 |
| INPUT 2: 0 | INPUT 6: 0 |
| INPUT 3: 0 | INPUT 7: 0 |
| INPUT 4: 0 | INPUT 8: 0 |
| NEXT --> | |

Figure 15 Analog Input Screen.

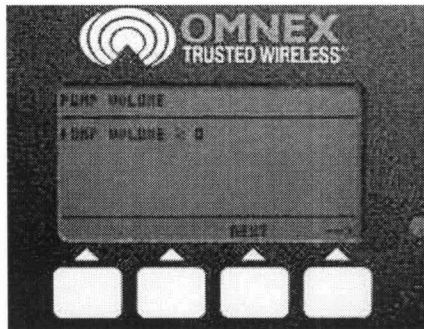


Figure 16 Analog Input Screen on T2600.

Joystick Screen

The joystick screen, shown in Figure 17 and Figure 18, show the raw joystick values.

| JOYSTICK INPUT | |
|----------------|------------|
| INPUT 1: 0 | INPUT 5: 0 |
| INPUT 2: 0 | INPUT 6: 0 |
| INPUT 3: 0 | INPUT 7: 0 |
| INPUT 4: 0 | INPUT 8: 0 |
| NEXT --> | |

Figure 17 Serial / Analog Input Screen.



Figure 18 Serial / Analog Input Screen.

1.4.4 Calibration Screen

Toggling the dedicated T2600 launched the calibration page. The application developer had the flexibility to choose whether a percentage text-based display or bar chart display would be more meaningful, and add the necessary code lines. (A 4 proportional R160 percentage text based output was included as a template for the default application. An alternative representation is shown in Figure 19) [4].

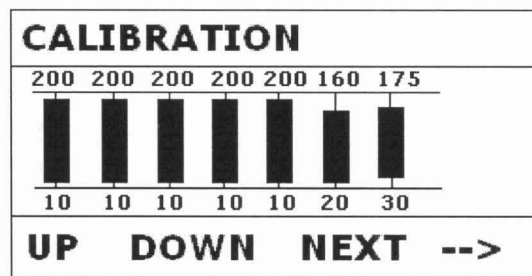


Figure 19 Calibration Screen.

Putting the T2600 in user calibration mode launched the user calibration screen.

1.4.5 System Screen

Additionally, there was a separate page defined with system information, including:

1. Battery Consumption

2. Stuck Switch
3. Tilt Status
4. Backlight On / Off [4]

Password for Factory (Production) Screens

The factory production screens were password protected as shown below in Figure 20 [4].

| |
|----------------------------|
| FACTORY PASSWORD |
| 0000 |
| UP DOWN NEXT --> |

Figure 20 Password Screen.

Backlight Screen

The backlight screen turned the backlight on and off (Figure 21) [4].

| |
|----------------------------|
| SYSTEM SETTINGS |
| BACKLIGHT 0 |
| UP DOWN NEXT --> |

Figure 21 System Settings Screen.

Key Map Definition and Handling

The application was written such that each screen had its unique keymap (Figure 22).

| |
|----------------------------|
| FACTORY PASSWORD |
| 0000 |
| UP DOWN NEXT --> |

Figure 22 Factory Password Screen.

CHAPTER 3: CONCLUSION

The LCD integration was successively implemented as outlined in the previous sections. The project progressed smoothly with no off-path diversions e.g. technical limitations of the transmitter board including e.g. insufficient flash memory. The display implementation received favourable feedback at the trade show demos.

At the time of this report, beta testing at a concrete-pumping customer's site is still in the not so distant future.

Future work on the project could possibly include:

- Allowing the bitmap tables to be uploaded from the receiver to ensure that the transmitter code is kept as simple as possible, allowing for greater complexity to reside on the receiver and the transmitter code to be as generic as possible across many applications.
- Adding support for daylight and temperature sensors.
- Development of a standard system menu to be accessed in a common way across many applications.

APPENDICES

Appendix A: ui_cfg.d Data Types

```
#define BUTTON_1 0x0001
#define BUTTON_2 0x0002
#define BUTTON_3 0x0004
#define BUTTON_4 0x0008
#define BUTTON_5 0x0010
#define BUTTON_6 0x0040
#define BUTTON_7 0x0020
#define BUTTON_9 0x0100

#define TEST 2

BEGIN_DATA
/*
    The following data descriptors describe data that are local to the
    user interface software. Do NOT alter the order of these descriptors
    without rebuilding the user interface software.
*/

DT_BYTE( lcd_bklt_bright, 0, 100, 1, "%3d", "%" )
DT_BYTE( lcd_contrast, 0, 100, 1, "%3d", "%" )

DT_SWORD( ui_temp, -100, 500, 10, "%3d.%1d", " C" )
DT_BIT_8( lcd_bklt_onoff, 0x01 )
DT_BIT_8( ui_daylite_onoff, 0x01 )
DT_STR( ui_version, 10 )
```

Appendix B: ui_cfg.d Enum Data Types

```
BEGIN_ENUM_TEXT_LIST(prop_control_list)
    // "01234567890123456789012345678901"
ENUM_TEXT_LIST_ITEM( prop_control_list_0, "" )
ENUM_TEXT_LIST_ITEM( prop_control_list_1, "BOOM 1" )
ENUM_TEXT_LIST_ITEM( prop_control_list_2, "BOOM 2 " )
ENUM_TEXT_LIST_ITEM( prop_control_list_3, "BOOM 3" )
ENUM_TEXT_LIST_ITEM( prop_control_list_4, "BOOM 4 " )
END_ENUM_TEXT_LIST( prop_control_list )
```

Appendix C: ui_cfg.d Bit Image

```
BIT_IMAGE_LIST(list_setting, blank, setting)
BIT_IMAGE_LIST(list_estop_act, estop_act, blank)
BIT_IMAGE_LIST(list_rabbit_mode, blank, rabbit_mode)
BIT_IMAGE_LIST(list_creep_mode, blank, creep_mode)
BIT_IMAGE_LIST(list_rd_conn, blank, rd_conn)
```

Appendix D: ui_cfg.d Bit Text

```
BIT_TEXT_LIST( on_off_list, "OFF", "ON" )  
BIT_TEXT_LIST( binary_list, "0", "1" )
```

Appendix E: ui_cfg.d Screen Definition

```
BEGIN_SCREEN( application, application, digital, joystick )

BUTTON( application_btn_0, BUTTON_1, UI_TBLS_ACT_NEXT_SCREEN, UI_TBLS_MD_NAVIGATE )
BUTTON( application_btn_1, BUTTON_7, UI_TBLS_ACT_PREV_SCREEN, UI_TBLS_MD_NAVIGATE )

TEXT( application_0, 1, 3, UI_TBLS_JX_LT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1, UI_TBLS_ZY_1,
"APPLICATION" )

TEXT( application_1, 1, 8, UI_TBLS_JX_LT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1, UI_TBLS_ZY_1,
"_____")

BIT_TEXT( application_6, 1, 17, UI_TBLS_JX_LT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1,
UI_TBLS_ZY_1, "PUMP ON", dig_pump_on, on_off_list )

ENUM_TEXT( application_key_0, 0, 58, UI_TBLS_JX_CT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1,
UI_TBLS_ZY_1, "", key_0, key_map_1 )

ENUM_TEXT( application_key_1, 40, 58, UI_TBLS_JX_CT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1,
UI_TBLS_ZY_1, "", key_1, key_map_2 )

ENUM_TEXT( application_key_2, 80, 58, UI_TBLS_JX_CT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1,
UI_TBLS_ZY_1, "", key_2, key_map_3 )

ENUM_TEXT( application_key_3, 120, 58, UI_TBLS_JX_CT, UI_TBLS_JY_TP, UI_TBLS_FID_4X6, UI_TBLS_ZX_1,
UI_TBLS_ZY_1, "", key_3, key_map_4 )

END_SCREEN( application )
```

REFERENCE LIST

- [1] OMNEX Control Systems, "Electronics in the Mobile Industry", pp. 1-21.
- [2] Peter Drexel, "External Interface Specification for the User Interface Subsystem of the T2300 Transmitter System", pp. 1-87, OMNEX Control Systems, February 2006.
- [3] Electronic Assembly, GMBH, EA GE128 – 6N 3V24 Graphic Unit 128 x 64 Data Sheet, pp. 1-12, February 1998.
- [4] Camille Jaggernaut, "T2600 LCD Specifications", pp. 1-21, OMNEX Control Systems, November 2007.
- [5] Zolly Skocic, "Building Tables for T2300 UI User Manual", pp. 1-17, OMNEX Control Systems, August 2006.
- [6] Zolly Skocic, "ORIGA Remote Data Exchange Protocol", pp. 1-16, OMNEX Control Systems, September 2005.
- [7] Heidi Lam, "Online Laboratory Results Reporting System", Simon Fraser University, pp. 22-26, 2006.