



ARL-MR-0933 • JULY 2016



# Design Report for Isolated RS-485 Bus Node

by Brian T Mays

Approved for public release; distribution unlimited.

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



# **Design Report for Isolated RS-485 Bus Node**

**by Brian T Mays**

*Sensors and Electron Devices Directorate, ARL*

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) July 2016		2. REPORT TYPE Memorandum Report		3. DATES COVERED (From - To) November 2015–April 2016	
4. TITLE AND SUBTITLE Design Report for Isolated RS-485 Bus Node				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Brian T Mays				5d. PROJECT NUMBER R.0019006.2.1	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-SES-S 2800 Powder Mill Road Adelphi, MD 20783-1138				8. PERFORMING ORGANIZATION REPORT NUMBER  ARL-MR-0933	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The Intelligence Surveillance and Reconnaissance Technology Integration Branch of the US Army Research Laboratory routinely develops custom embedded hardware for use in distributed command and control systems. This report covers the design details of a smartphone-controlled wired RS-485 network. The Android-based smartphone or tablet is used in conjunction with a USB to serial bridge to operate as the bus master in the system. The Android device operates in USB Host mode and communicates to the RS-485 bus as if a single peripheral on the USB bus.					
15. SUBJECT TERMS Android, RS-485, isolated, USB, smartphone					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UU	18. NUMBER OF PAGES  28	19a. NAME OF RESPONSIBLE PERSON Brian T Mays
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-2185

## Contents

---

<b>List of Figures</b>	<b>iv</b>
<b>Acknowledgments</b>	<b>v</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Background	1
1.2 Master Control Node	2
<b>2. Isolated Node Daughter Card</b>	<b>2</b>
2.1 Theory of Operation	2
2.2 Electrical Isolation	3
2.3 Local Microprocessor	4
<b>3. Android Control Software</b>	<b>4</b>
<b>4. Conclusions and Future Developments</b>	<b>6</b>
<b>Appendix A. Schematics</b>	<b>7</b>
<b>Appendix B. STSLED Software Quick Start Guide</b>	<b>11</b>
<b>STSLED Software Quick Start Guide</b>	Error! Bookmark not defined.
<b>List of Symbols, Abbreviations, and Acronyms</b>	<b>19</b>
<b>Distribution List</b>	<b>20</b>

## List of Figures

---

Fig. 1	System topology (485P and 485N twisted pair with each end terminated in 120 ohm).....	1
Fig. 2	Isolated Node Daughter Card (INDC) .....	2
Fig. 3	INDC isolated ground regions .....	4
Fig. 4	Software components and dataflow .....	6
Fig. A-1	Main power and interface schematic .....	8
Fig. A-2	Local processor and RS-485 schematic .....	9
Fig. A-3	Isolated digital and analog input/output schematic.....	10
Fig. B-1	Main screen .....	12
Fig. B-2	Main menu view .....	13
Fig. B-3	Configuration screen .....	14
Fig. B-4	Discover screen .....	15
Fig. B-5	Assigning unit serial number .....	16
Fig. B-6	Normal operation .....	17

## Acknowledgments

---

Mark Ware of the US Army Research Laboratory is acknowledged for his contribution of the printed circuit board layout of the Isolated Node Daughter Card.

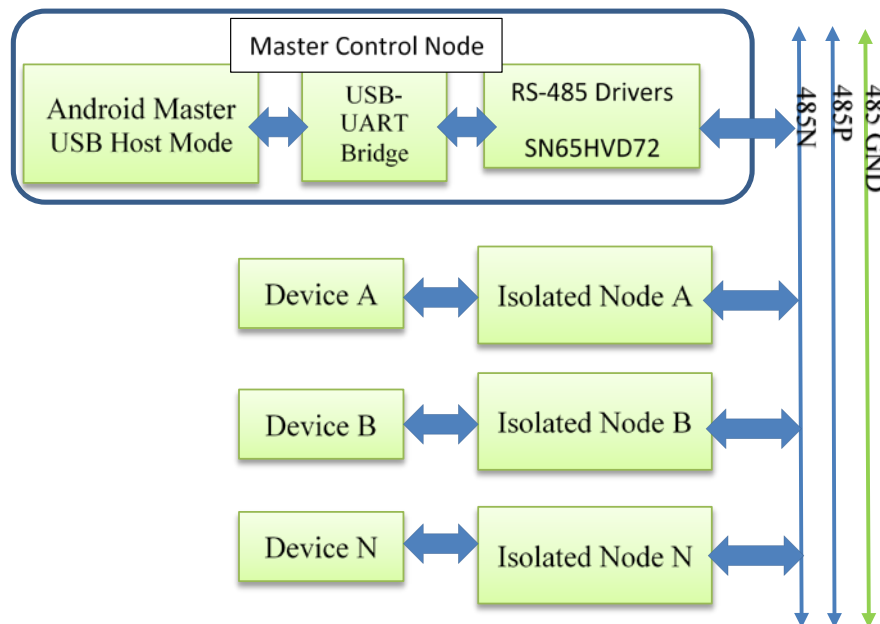
INTENTIONALLY LEFT BLANK.



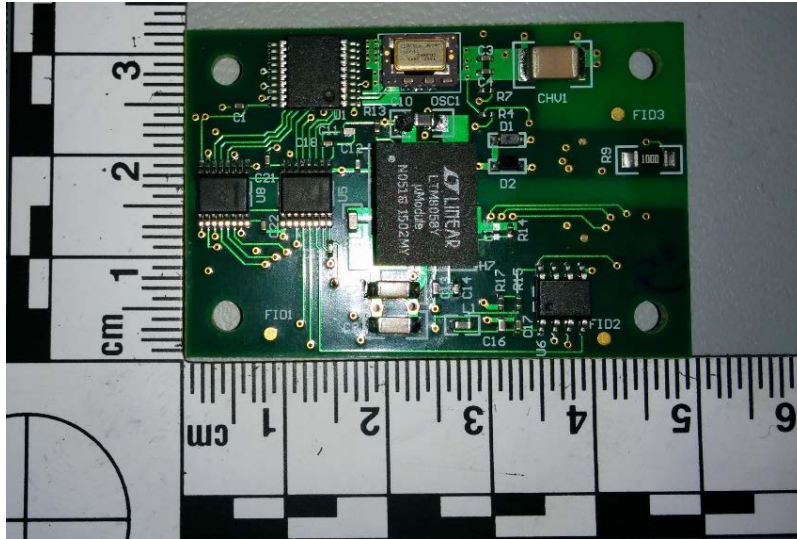
# 1. Introduction

## 1.1 Background

The Intelligence Surveillance and Reconnaissance (ISR) Technology Integration Branch of the US Army Research Laboratory (ARL) routinely develops custom embedded hardware for use in distributed command and control systems. This report covers the design details of a smartphone-controlled wired RS-485 network. The topology of the system is shown in Fig. 1, in which an Android-based smartphone or tablet is used in conjunction with a USB to serial bridge to operate as the bus master in the system. The Android device operates in USB Host mode and communicates to the RS-485 bus as a single peripheral on the USB. In general applications this topology allows the Android, with a hub attached, to communicate with multiple USB peripherals for additional data collection and fusion. The RS-485 endpoints are slave devices to the Android device but are isolated from the complexities of the external USB. Each of the endpoints contains an Isolated Node Daughter Card (INDC), shown in Fig. 2, which was designed to be a reusable component for all end nodes.



**Fig. 1** System topology (485P and 485N twisted pair with each end terminated in 120 ohm)



**Fig. 2 Isolated Node Daughter Card (INDC)**

## 1.2 Master Control Node

---

The master control node was implemented with a USB to universal asynchronous receiver/transmitter (UART) bridge chip produced by Future Technology Devices International (FTDI). While several integrated circuit (IC) manufacturers produce USB-to-UART serial converters with similar features, the FTDI IC was selected for software driver support across various operating systems. Vendor-supplied Android support was desired without the requirement of building a custom Android kernel. Some manufacturers claim native Android support but integrate their Linux drivers into the Android kernel and do not use the native USB libraries in Android. This is problematic for deployment, as a custom “rooted” device must be supplied to the end user versus simply installing an application (app) on the end user’s target device.

## 2. Isolated Node Daughter Card

---

### 2.1 Theory of Operation

---

The INDC is designed to be a reprogrammable subcomponent to an endpoint on the RS-485 bus. This feature allows the communication protocols and subsequent action performed by the INDC to be customized per mission requirements. The initial use of the INDC in the system depicted in Fig. 1 has all the INDCs operating in a slave mode with an Android device operating as the master. There is no restriction in using this topology; the INDCs could be programmed to operate in peer-to-peer or self-contained master slave topology. The initial use for the modules

was to support engineering tests. Using the Android device as the master allowed for human control, high data visibility, and data logging. In an embedded closed-loop application, the tablet would be replaced with a rugged controller or potentially one of the INDCs operating as the master.

For the system depicted in Fig. 1, a custom binary message protocol was used for efficiency. The over-the-wire protocol uses a positive acknowledgment handshake between the masters and slaves so that all communications traffic is monitored for integrity and reliability. All of the command and data packets have variable length in the format specified in the following list, with the minimum packet size of 10 bytes (no payload data), and include a checksum for monitoring bit errors and possible bus collisions.

- 2-byte start of packet, defined as 0x424D
- 1-byte destination address (DES), 0xFF reserved for broadcast messages
- 1-byte source address
- 2-byte message identification, big endian
- 2-byte payload length (MPL), little endian
- N-byte payload (MPL: bytes of data)
- A 2-byte checksum is produced with the 8-bit Fletcher algorithm as specified by RFC 1145 and calculated over entire packet starting at DES.

## 2.2 Electrical Isolation

---

The electrical isolation boundaries of the INDC are indicated in Fig. 3 by the dashed green lines. Four distinct ground regions exist in the design: device, node, bus, and chassis. The device region ground is common to the endpoint hardware, which is being controlled by the INDC. The devices also provide the power for the INDC, allowing endpoints to run at varied local voltages while maintaining an operating bus voltage of 3.3 V across all nodes. This also simplifies the interconnect harness by requiring that only communication lines be present. The chassis region is solely used to shunt noise transient from the transient voltage suppression diodes on the RS-485 signal lines. The noise path is shown in red as a minimum layout requirement on the printed circuit board design and is intended to be connected to a protective earth ground (case) and not the power ground of the endpoint device. The node region is the isolated ground for all the internal INDC electronics. The final region is the bus region, which is used to pin the grounds of multiple INDCs to common voltage. This was done to handle cases where damaging static voltages

could build up between INDCs. If no bleed path was provided to equalize, damaging voltage potentials across the RS-485 bus could develop. As the RS-485 signals are differential, no steady-state current should flow through the R100 100-ohm pinning resistor. The schematics are provided in Appendix A.

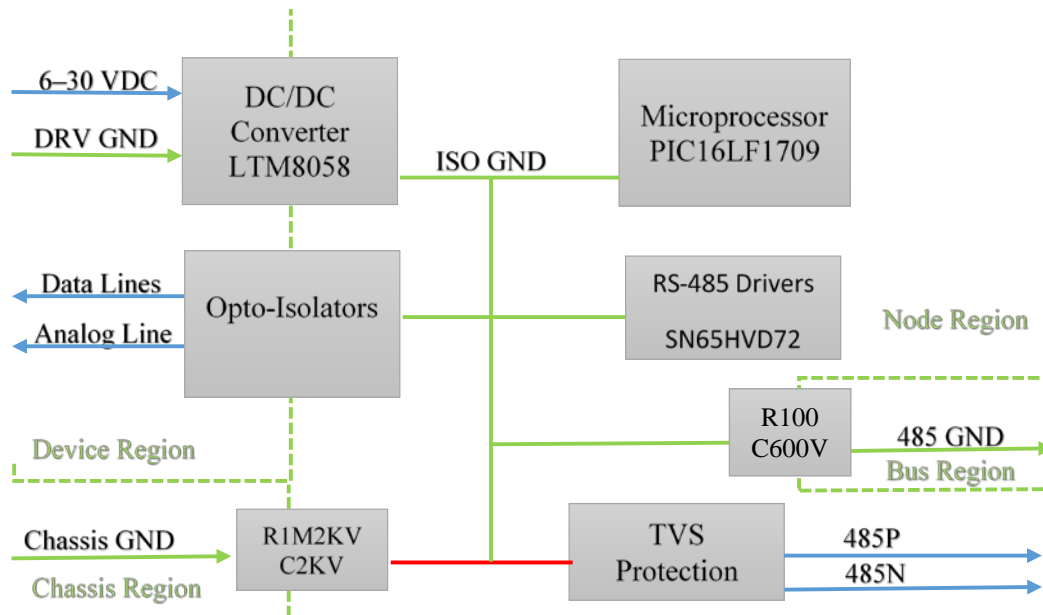


Fig. 3 INDC isolated ground regions

## 2.3 Local Microprocessor

A local microprocessor was included on the INDC to allow the endpoints to operate as intelligent nodes and minimize communication traffic. A low-power processor was selected from Microchip to support custom “house” logic for various node applications. In general, only high-level commands need to be sent to the nodes, and the local processor can execute a sequence of events with the endpoint device and communicate back to the bus master as required. The micro is responsible for parsing the over-the-bus protocol and responding to the master commands.

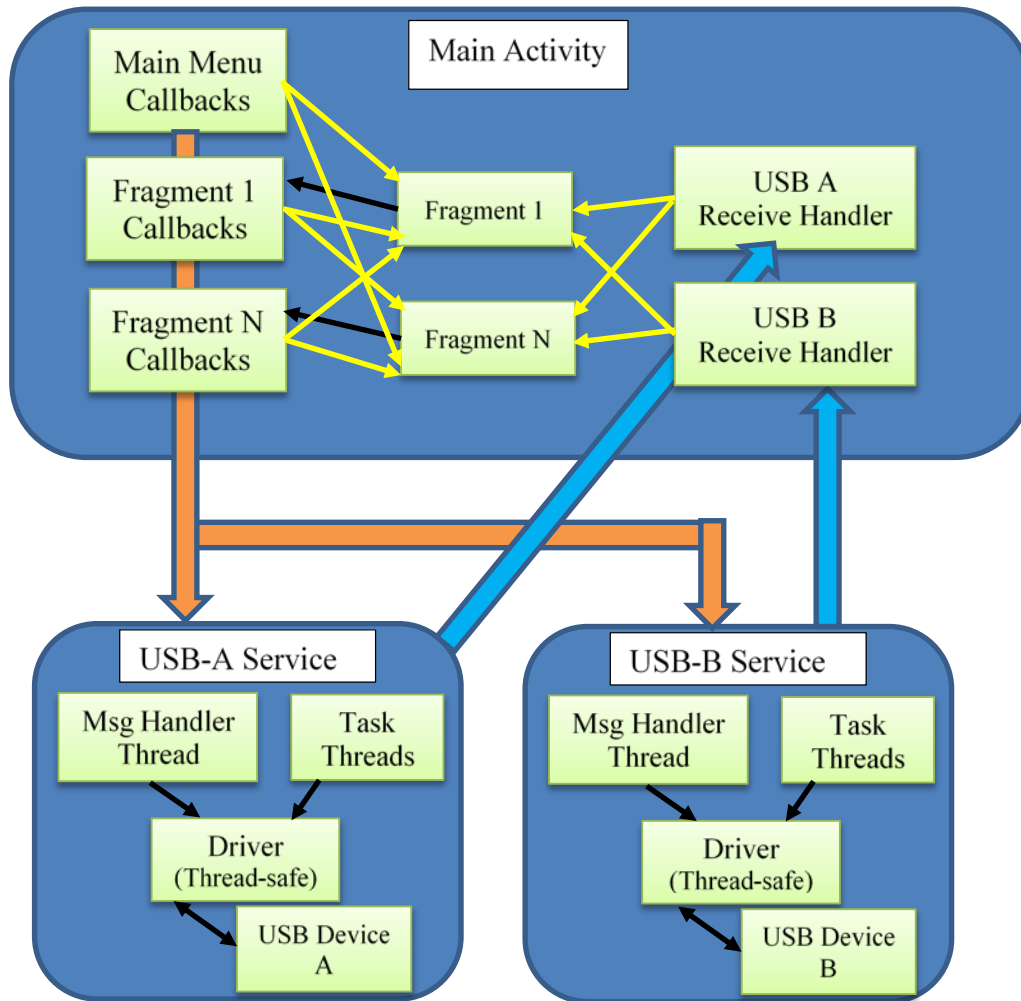
## 3. Android Control Software

As part of the first development effort to integrate the INDC into a complete system, the Android-based control application STSLED was developed by ARL. A multipane fragment-based application was developed to provide an engineering control application of multiple USB clients, one of which is was the INDC. The basic software architecture provides an Android-bound service component for each of the USB clients. These service components implement a handler thread that receives messages from the main activity and processes the requests. The service

spawns any worker threads required to complete long-duration tasks in the background. This structure isolates the main application from the low-level device details and further isolates the services from each other. The main activity implements a receive handler for each of the services to reply to when a command has been completed.

The multipane context of the application is implemented with Android fragments. The main activity has sole responsibility for managing which fragments are visible and for all fragment-to-fragment information exchanges. This is accomplished by each fragment having its own internal callback interface and requiring that the main activity implements all interfaces for the fragments it hosts. This allows the main activity to be the sole location for house logic on how a specific fragment's information may be used to notify another fragment that some action is required. This isolation also allows for greater reuse of the fragments in subsequent projects.

Figure 4 shows the software components logical association and communications paths. The blue boxes are the main thread groups: Main Activity, which is the UI thread, USB-A, and B Service collection of threads, respectively. The green and blue arrows indicate Android messages that are passed between threads and handled by the operating system in a thread-safe manner. The green arrows are commands being passes to the required services for action and the blue arrows are responses to the command request being passed back to the Main Activity for processing. Within an individual service, only the device driver code library communicates to the USB device and is required to be thread-safe. The dataflow in the main activity is restricted to ensure isolation between components by requiring no direct fragment to fragment interactions occur. Only the fragment callbacks in the Main Activity can receive information from one fragment notifying another fragment of required action through the fragment's public methods. The next isolation boundary is that only the callbacks in the main activity can generate commands to the services. Also, any returning information to the main activity is processed by the receive handlers, which in turn determine which fragments need to be notified. The yellow arrows indicate fragment manipulation via calls to public methods. The black arrows in the main activity indicate the fragments requesting action from the main activity via a predefined internal interface of the fragment.



**Fig. 4 Software components and dataflow**

It is beyond scope of this report to cover the entire theory of operation for the STSLED application in this report, but the Quick Start Guide is included in Appendix B as reference material.

## 4. Conclusions and Future Developments

The INDCs have been successfully integrated by engineers of the ISR Technology Integration Branch into a distributed multidrop RS-485 network. The system was successfully used on a tactical vehicle in a high electromagnetic interference environment. The implementation of the INDC as a daughter card for endpoint devices ensured reusability for future applications. The use of an Android-based smartphone as master controller proved to be a very rugged and simple device to operate by test engineers due to familiarity with standard mobile applications used in daily life.

## **Appendix A. Schematics**

---

---

The schematics in this appendix appear in their original form, without editorial change.

Approved for public release; distribution is unlimited.

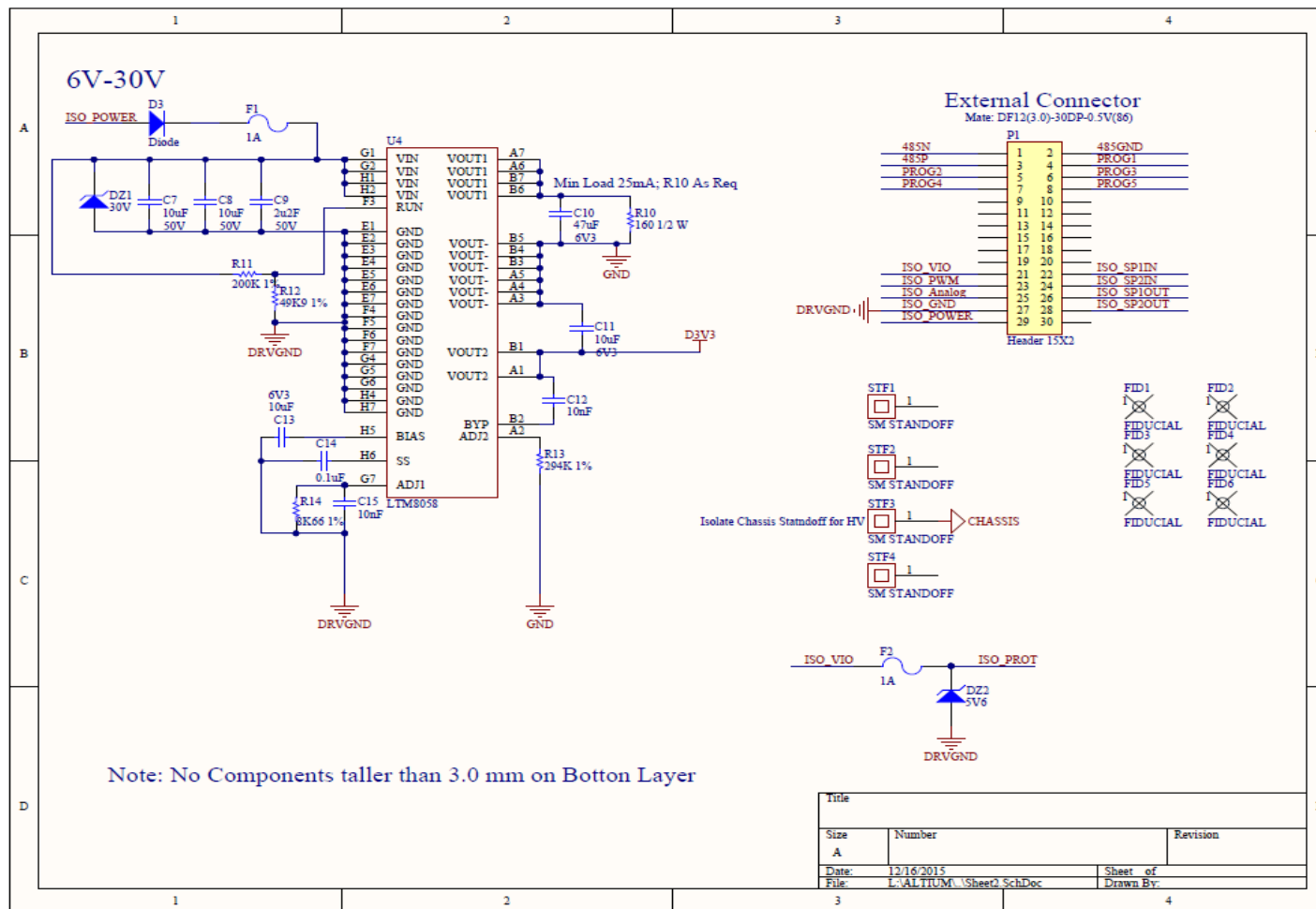


Fig. A-1 Main power and interface schematic





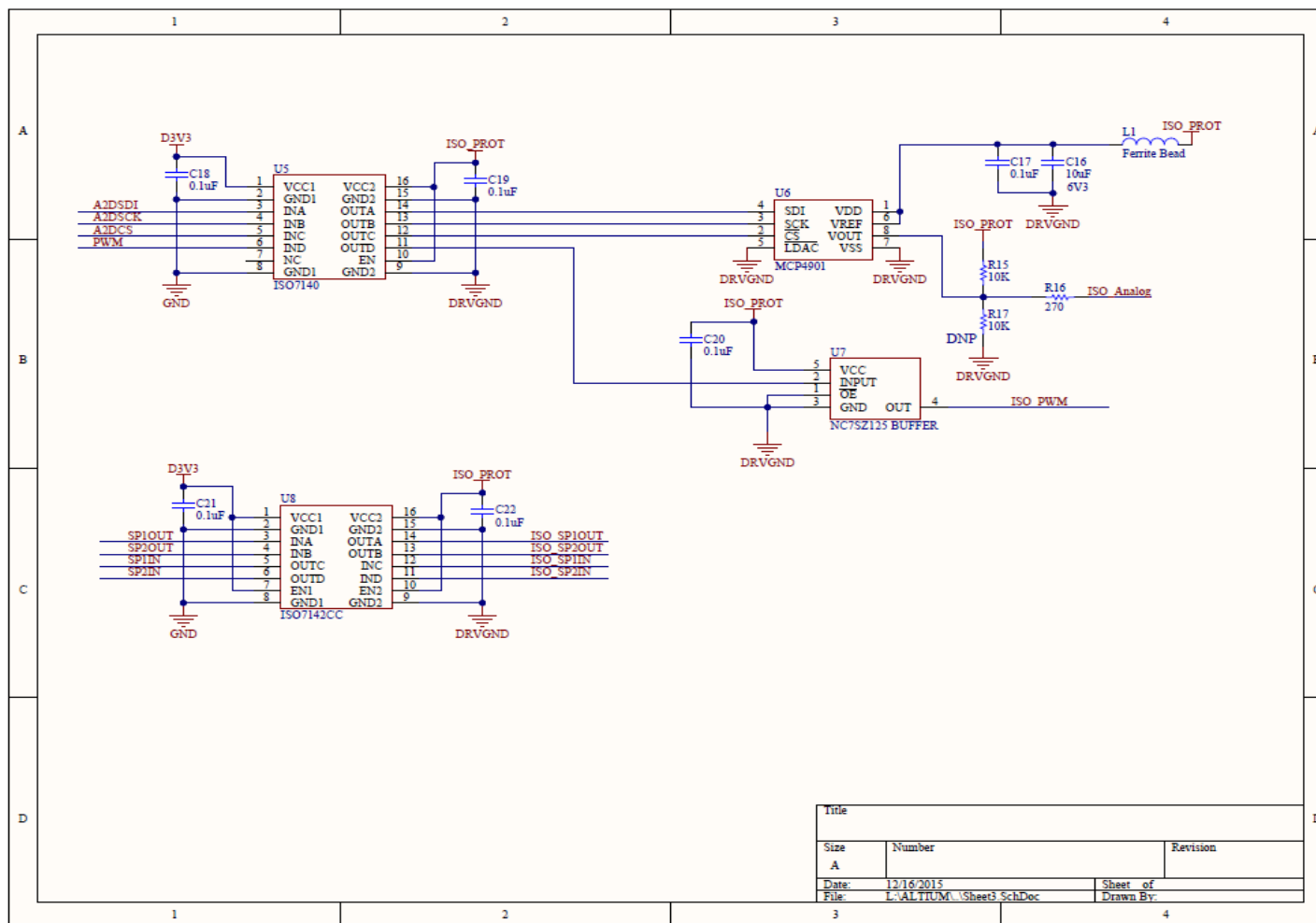


Fig. A-3 Isolated digital and analog input/output schematic

## **Appendix B. STSLED Software Quick Start Guide**

---

## Background

The main control screen of the Android-based control application STSLED, developed by the US Army Research Laboratory, is shown Fig. B-1 with primary user controls and indicators pointed out. From this screen all primary system command and control can be performed. Two setup screens used for discovery and configuration are navigated to through the Menu button in the top right of the application. Tapping the Menu button drops down the menu items, as shown in Fig. B-2.

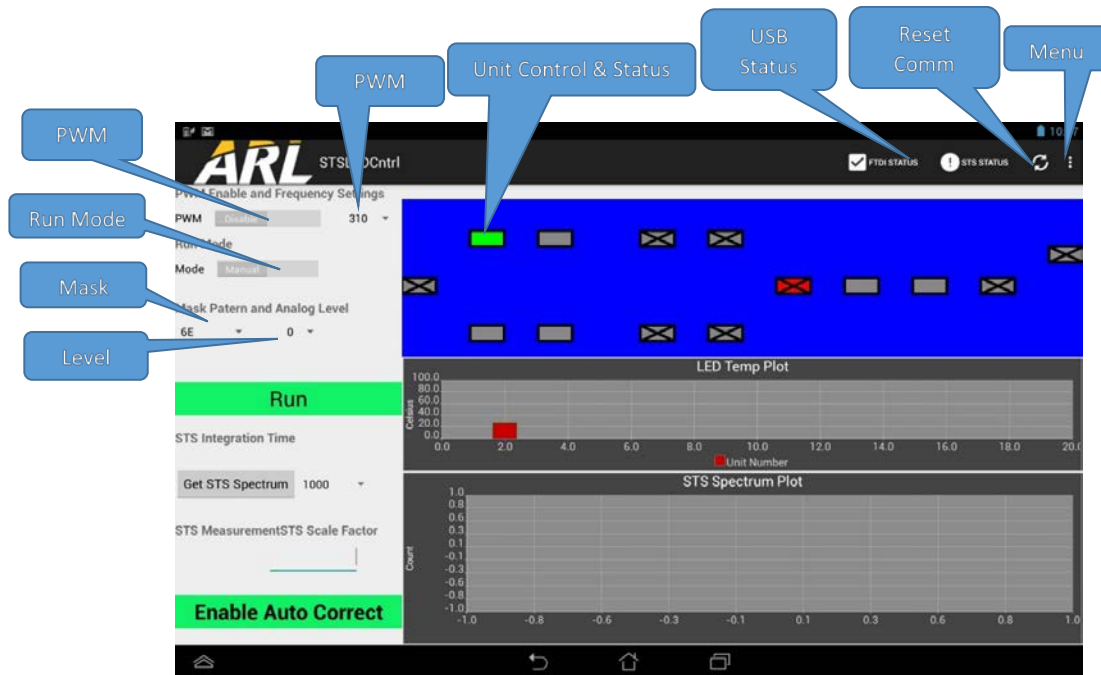
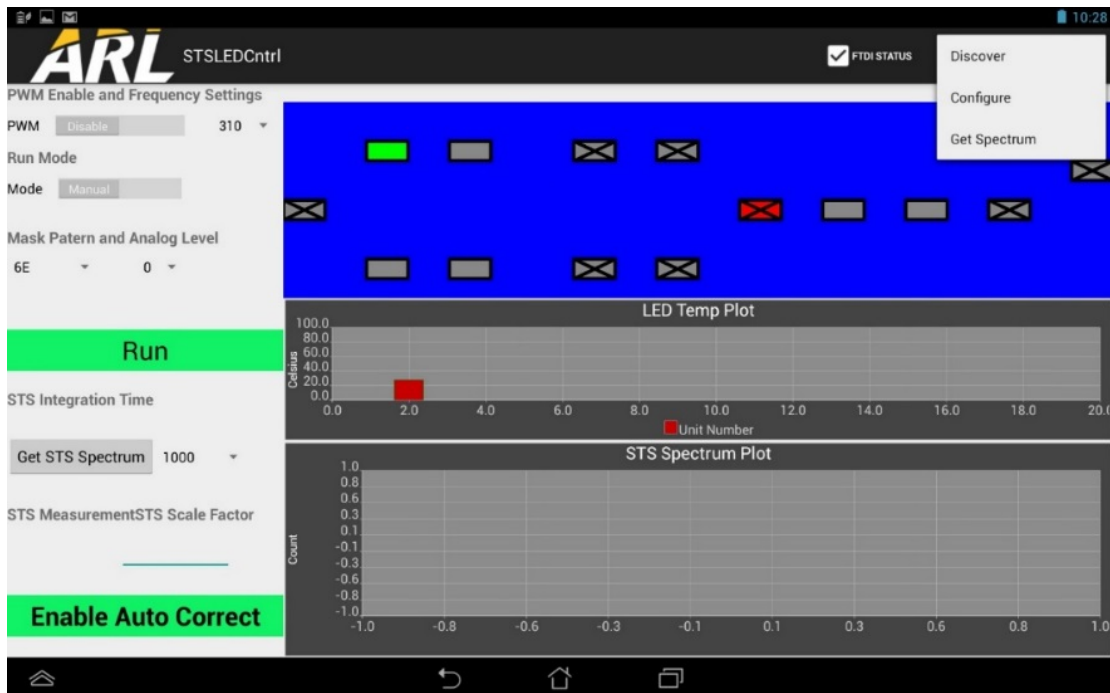


Fig. B-1 Main screen



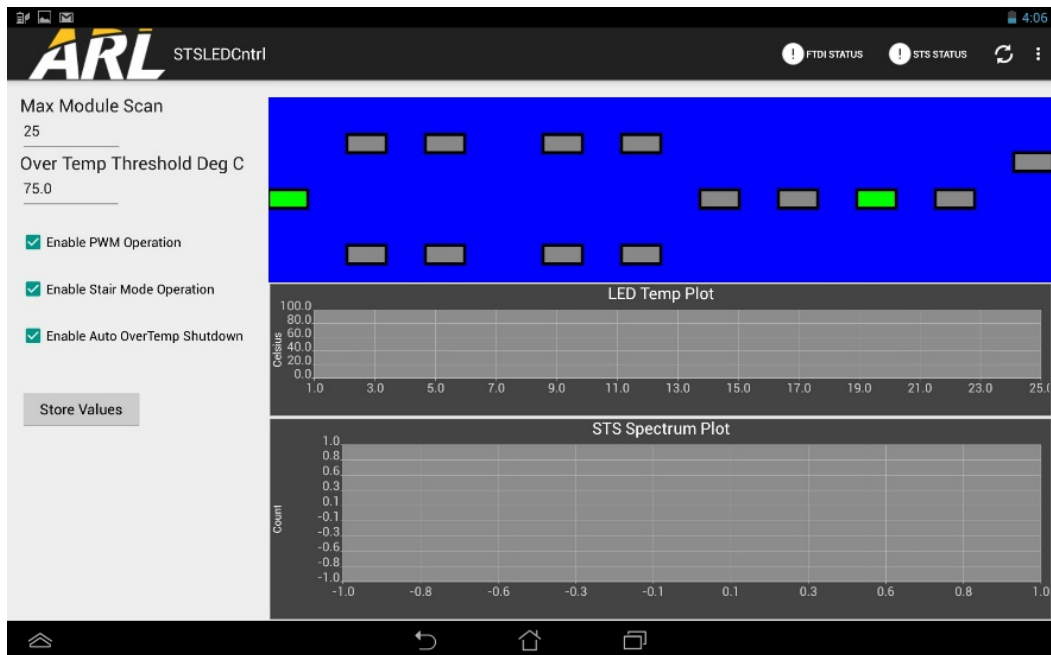
**Fig. B-2 Main menu view**

## Initial Setup

The system setup procedure is required once at initial installation and anytime units are swapped out or replaced. The function of the setup phase is to tell the software where each unit is installed logically by serial number. After the physical installation, note the serial number and corresponding physical locations for the units.

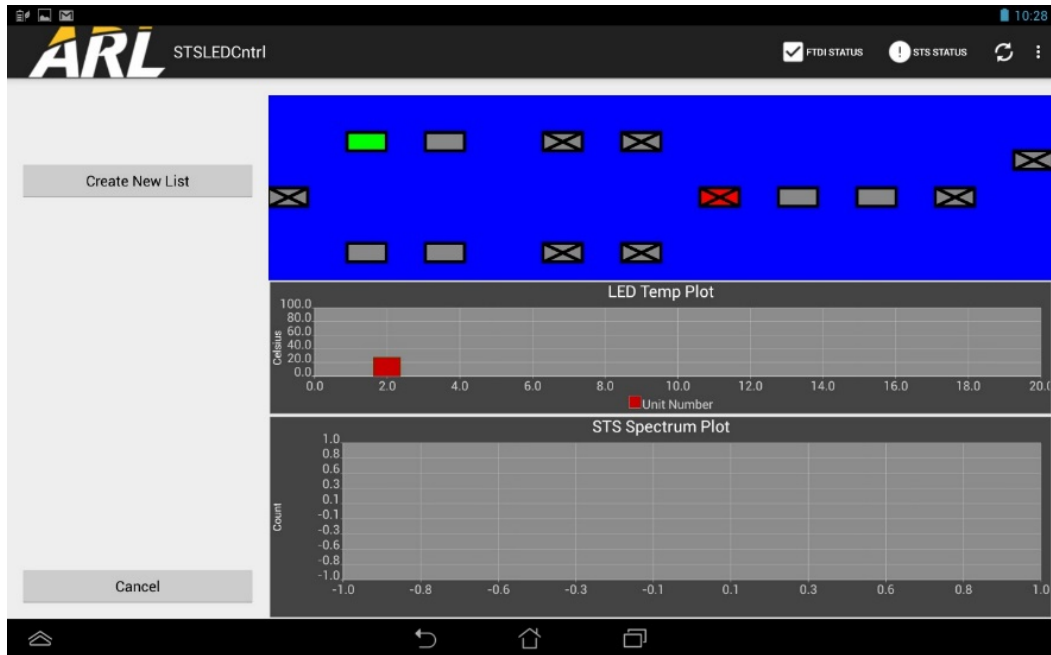
The software is used to first scan the bus to determine what units are on the bus by serial number. The user then assigns a serial number to each location on the main screen. This is accomplished in the following 3-step process:

1. Open the Configure screen via the Menu button on the main screen. The Configure screen should now be seen as in Fig. B-3. Enter the Max Module to Scan parameter, which should be at least the highest serial number installed. Tap the Store Values button to save and return to main screen.



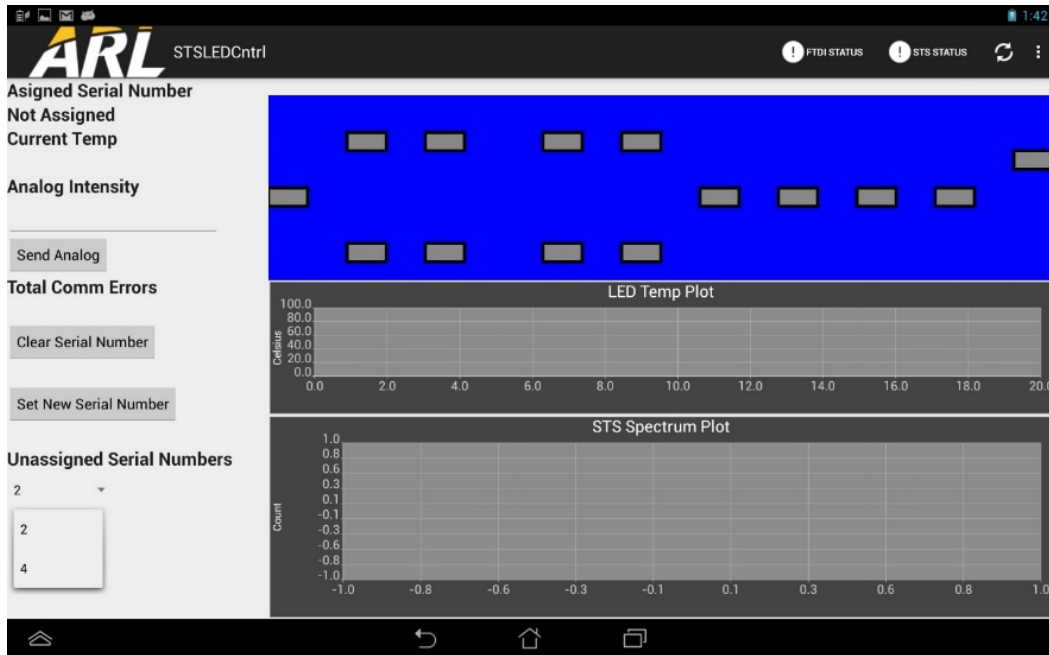
**Fig. B-3 Configuration screen**

2. Open the Discovery screen with Menu button on the main screen. The Discovery screen should now be seen as in Fig. B-4. Tap the Create New List button to begin the scanning process. The view will now return to the main screen but the user must wait until an indicator message appears stating that discovery is complete and specifying the number of units found on the bus. This process can take 2 s/module scanned to completion.



**Fig. B-4 Discover screen**

3. The user must now assign serial numbers to the physical location on the unit map. This is done by selecting each of the unit control boxes one at a time and setting a serial number for that unit. Once the specific unit control is tapped, the unit control screen appears, allowing the serial number to be selected from a pull-down list and the number to be assigned with the Set New Serial Number button, as shown in Fig. B-5. The Clear Serial Number button can be used to unassign a unit in case of entry error. Repeat for all installed locations.



**Fig. B-5 Assigning unit serial number**

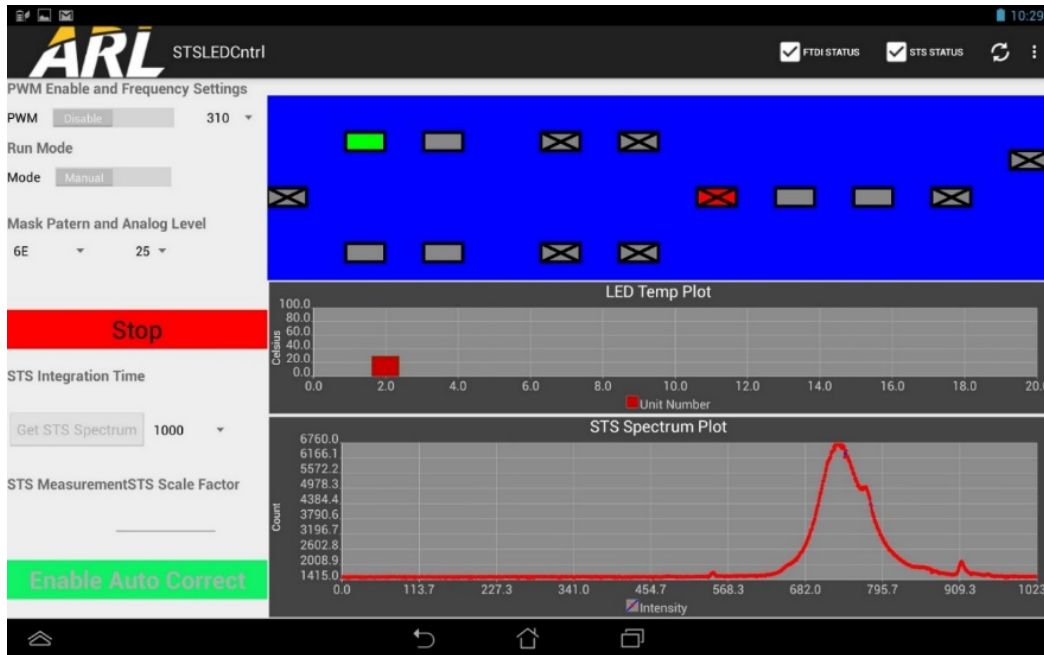
## General Operation

Normal operation of the system is accomplished on the Main Screen level. From there, the user will first select the various run configuration options that are highlighted in Fig. B-1. The user selectable options include the following:

- Toggle to enable/disable pulse-width modulation (PWM) operation.
- Use drop-down to select PWM frequency.
- Toggle to select Manual/Stair mode operation.
- Mask pattern to apply to the next test case.
- Use drop-down to select Analog level.

Once all configuration parameters are entered for a specific test, the Run button is used to start the test. The Run button now turns into a Stop button. In Manual run mode, the test continues until the user taps Stop. In Stair mode, the Analog level is stepped 0%–100%. During this period the Stop button is grayed out as the user cannot abort this operation. An example of operating in Run mode is shown in Fig. B-6.





**Fig. B-6 Normal operation**

Individual unit status is represented by the following colors and hash marks:

- Green is unit assigned and normal operation.
- Red is unit assigned and communication errors have occurred.
- Yellow indicates that the unit is in over temperature condition and test should be suspended.
- Gray indicates that the unit has not been assigned a serial number.
- Hash marks indicate that Mask pattern in use: X = unit off, / = unit at partial power, and blank = 100 %.

## File Storage

While the application is in run mode and the STSLED is connected, data from the STS is retrieved and stored at the specified Integration rate. The files are stored on internal storage in the Download directory. The filename contains the date and time stamp at the start of the run and is Excel-compatible with the extension “.csv”. The files in this directory must be manually deleted from the directory either from local File Manager App or while connected to host personal computer.

It is critical that the system date and time be set prior to operation to ensure filenames are correctly time-stamped.

## **Shutdown**

---

Normal shutdown is achieved by using the Android back navigation button at the bottom of the screen. If the App is nonresponsive, the Android back stack button on the bottom right can be used to first put the app on the system back stack and then swipe it off to fully destroy it.

## **Recovery**

---

To fully restart the system, both the Android app should be restarted and the remote units rebooted. This is accomplished as follows:

1. Unplug the universal serial bus (USB) cable from the tablet.
2. Go to Android back stack via button on bottom right and swipe off STSLEDCntrl.
3. Cycle power to remote units with On/Off switch at base hub.
4. Reconnect USB cable and the app should relaunch.
5. Confirm USB status indicators.

## List of Symbols, Abbreviations, and Acronyms

---

app	application
ARL	US Army Research Laboratory
DES	destination address
FTDI	Future Technology Devices International
GND	ground
IBuN	isolated bus node
IC	integrated circuit
INDC	isolated node daughter card
ISR	intelligence surveillance and reconnaissance
MPL	message payload
PWM	pulse-width modulation
UART	universal asynchronous receiver/transmitter
USB	universal serial bus

1 DEFENSE TECHNICAL  
(PDF) INFORMATION CTR  
DTIC OCA

2 DIRECTOR  
(PDF) US ARMY RESEARCH LAB  
RDRL CIO L  
IMAL HRA MAIL & RECORDS  
MGMT

1 GOVT PRINTG OFC  
(PDF) A MALHOTRA

12 DIR USARL  
(PDF) RDRL SES A  
B LISS  
T WALKER  
RDRL SES P  
C REIFF  
H VU  
D HULL  
M SCANLON  
S TENNEY  
RDRL SES S  
B MAYS  
M WARE  
B MARY  
D WARD  
M YOUNG