Getting Started with Sender V4

This is a quick start guide. For detailed information of the shell commands, variables, and scripts, see Shell.doc.

Platform

The platform is an off-the-shelf STMicrosystems Nucleo-F429ZI along with a custom daughter board. Some of the functionality can be explored without the daughter board, but the real time clock battery, SD card, and the external interfaces will not be functional. There is an existing revision of the daughter board and *could* be built, a new version will be out soon, so it makes sense to wait for the new version.

Prepare the SD Card

Copy the contents of the Card Contents folder to the SD card and insert it into the socket.

Development Serial Connection

Connect a micro USB cable from CN1 (the USB connector on the J-Link extension) to the PC. Run a terminal emulator, like TeraTerm, and set the com parameters to 115200 N,8,1.

This port will allow communicating via serial, debugging using J-Link, and firmware updates (MBed). The USB port on the opposite end, CN13, can be used as a mass storage device and will eventually be used for a serial connection as well. Once this occurs, this will become the main USB serial connection.

On the terminal emulator press Enter and verify that the connection and port settings are correct by the prompt "/>". This is the shell, and interaction with the sender will start here.

Type "dir <Enter>". This will list the files on the SD card.

```
Directory of /
Name            Size      Modified Date - Time  Attributes
SEND            <dir>      1/01/2000   7:25:20  ----D-
TEST.SCP        175       30/12/2019  16:42:34  -----A
RESULTS         <dir>      0/00/1980   0:00:00  R---D-
CONFIG.INI      142       24/12/2019   0:35:36  -----A
TESTS           <dir>      9/11/2019   9:04:56  ----D-
HEADER.TXT      1604       1/01/2000  16:38:56  -----A
TEST.LOG        0         22/02/2020  17:16:32  -----A
WEB             <dir>      2/10/2019  22:40:10  ----D-
IDLE.PKT        66        23/09/2019  22:36:50  -----A
CLOCK.SCP       173        5/04/2020  20:04:18  -----A
STARTUP.SCP     27         6/04/2020  20:35:36  -----A
```

Type "help" or "?". This will list the visible commands, system variables, and scripts.

```
Commands:
help          clrscr        clreol        gotoxy        cursor        color
variables     echo          led           script        dir           type
```

```
del          md           cd           atrib        copy         tasks
cab          loco         train        disp         write        read
send         bittest      sendzero     sendone      scopea       scopeb
warble       stretched    reset        hard         idle         packet
ymodem

Variables:
version      serial       time         date         timefmt      datefmt
ipaddress    ipmask       gwaddress    dhcp         port         track
path

Scripts:
TEST.SCP     LS.SCP       CLOCK.SCP    STARTUP2.SCP
```

If you type “help -c <Enter> the output will be colorized by category type.
If you type “help -a <Enter>. All the commands, even the hidden commands will be listed.
Type “help <name> <Enter>” for a reminder string for any command, variable, or script (if the script file has a comment as the first line),

Type “version <Enter>”. The current value of the variable ‘version’ will be output. This variable is read-only, but other variables may be set by typing the variable name followed by the value.
To set the time and date, type “time hh:mm <Enter>”, and “date mm/dd/yyyy <Enter>”. The order of the date elements follows the date format ‘datefmt’ variable.

Type “ls”. The script ‘LS.SCP’ will execute. This is a simple script to emulate the Unix style directory command, albeit in color.
Take a look at that script by typing “type ls,scp” and the following will display:

```
rem linux style directoy - now in color
dir -c %1
prompt
```

The first line is a comment. A comment can be anywhere in a script but has special meaning if it is the first line. If you type “help ls <Enter>”, that line will display.
The following lines are any command, variable, or script. In this case the ‘dir’ command with two arguments. The ‘-c’ displays the output in color, and the %1 argument substitutes an argument included with the script invocation, such that “ls results <Enter>”, will display the contents of the ‘results’ directory.
The last line outputs the prompt string and is a nice way to finish a script.

Take a look at the test.scp script. Run it and the view it. In it you will see some of the ANSI commands along with the delay function.

Scripts run in the background and more than one script can run at the same time. Run clock.scp. This script loops forever (well, almost. It won’t be running the next time you power it up and you can kill it). It has ANSI cursor positioning commands that will display the time and date in the upper right corner of the screen (if using the teraterm’s default screen size). Type “script clock kill <Enter>” to stop it.

As of this writing, the clock script randomly crashes.

Decoder Testing

The ‘send’ command behaves the same as the V3 send command did. It is documented in XXXX.doc. There is a fully automated suite of tests along with a manual mode for specific packets and other commands. Some of these commands are also available in the shell. The shell commands different slightly from the Send-Manual commands; none of them repeat automatically (except the sendone and sendzero commands). Without a count argument, the packet will only be sent once.

Command	Description
sendzero	Send DCC 0 Pattern (continuous)
sendone	Send DCC 1 Pattern (continuous)
scopea	Send Scope A Patterns
scopeb	Send Scope B Patterns
warble	Send a Warble Pattern
stretched	Send Stretched 0 Pattern
reset	Send Normal Reset Packets
hard	Send Hard Reset Packets
idle	Send DCC Idle Packet
packet	Sends a packet fully described is a PKT file

packet

The ‘packet’ command is intended to give decoder developers a consistent method to produce packets with any imaginable bit timing, including packets that are not DCC. The bit description is driven from a file, so a library of different packets may be developed. A series of packets can then be used in a script.

Here is the content of the idle.pkt file:

```
rem a text stream that will be displayed using help
preambles = 18
200, 100, 1
116, 58, 8
200, 100, 10
116, 58, 9
```

The first line may have a rem statement to display using help, The packet definition starts with the word “preambles” followed by an equal sign and the count. If the preamble count is zero, no preambles will output and the packet will be defined by the following descriptions.
There can be many lines following that have following format:
[period], [pulse], [count]
The period and pulse are specified in microseconds. The period is the width of both the high and low portion of the bit and the pulse is the width of the first half of the bit. The count is the number of times the bit is repeated, 0 is the same as 1.

Telnet and Web

As of this writing, the Ethernet port will connect, but the Telnet and Web Server apps are not complete. Telnet will connect to the shell in the same manor as the USB serial connections. The Web Server may also connect to the shell (if we see that is it useful) and be able to display documentation on a web browser.

Command Station

The Sender contains a full featured Command Station based on the Wangrow-SystemOne / NCE Cab protocol. The Lenz XpressNet protocol is possible, but the display on the NCE Cab provides added feedback. The Cab connection is shared with the LCC connection on the same connector. An adapter or breakout harness is necessary to connect to a Cab.