Low Cost DCC Controller with Service Mode Programming

Serial/Bluetooth & 2amp/10amp Version 1.7

December 11 2017

Disclaimer

Provided in this document are the details of how to build a very low cost 2amp dual track DCC controller or a one track 10amp controller. The hardware design is free. Software that resides in the ARM Cortex M3/M4 that controls the DCC track lines and associated Windows and Android software is purchased from eBay.

The designers accept no responsibility for any damage to any train or accessory decoder connected to this DCC system through incorrect assembly or use of the hardware design.

Please read s-9.1_electrical_standards_2006.pdf NMRA standard before purchasing and using a power supply. Also note some cheap power supplies can give over voltage output.

Please read this document completely before assembling the controller or purchasing the software.

Included at the end of this document is a list of decoders know to work with this DCC system. This list will increase with time as more and more people start using this low cost controller for service mode programming and layout control.

Please let us know of any decoders not listed that are working with the DCC control system.

Contents

Introduction

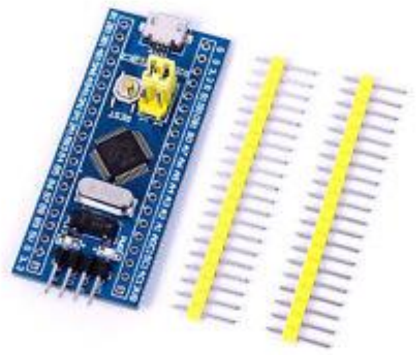
This book describes how to build various low cost DCC controllers. The controllers all support multiple connections via Bluetooth or serial to either Windows PCs or Android phones or tablets. The connections allow either multiple users to control different trains on a layout or wireless and wired controllers to be used by one person (walk around controller) on a layout.

The DCC controller is a low cost modular design that requires little or no soldering to build and consists of between four and six components depending on configuration. All of the components are readily available through eBay and other internet outlets.

All hardware designs have common Windows PC software and common Android phone/tablet applications.

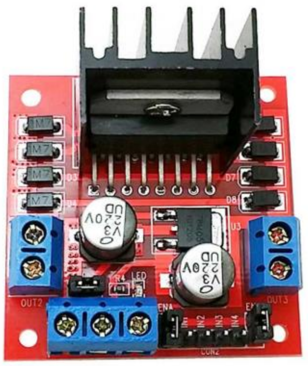
The DCC track control software runs on either an STM32F411RE Nucleo board (no soldering required) or on an STM32F103 Arduino Nano board which requires some soldering depending on which board is purchased.

STM32F411RE

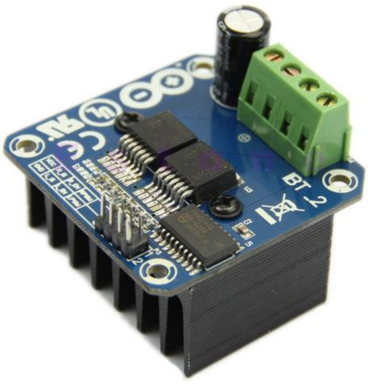


STM32F103

There are also two choices of current drive capability, a 2amp dual track version and a single track 10amp version. The dual track 2amp version uses an L298N H Bridge, the 10amp version uses an IBT-2 H Bridge.



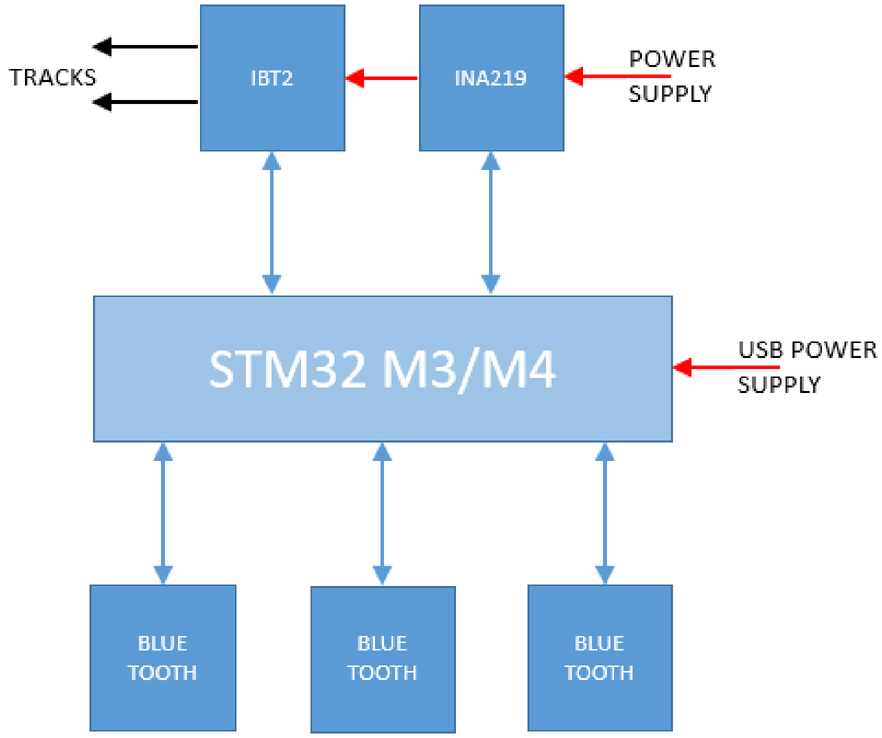
L298N



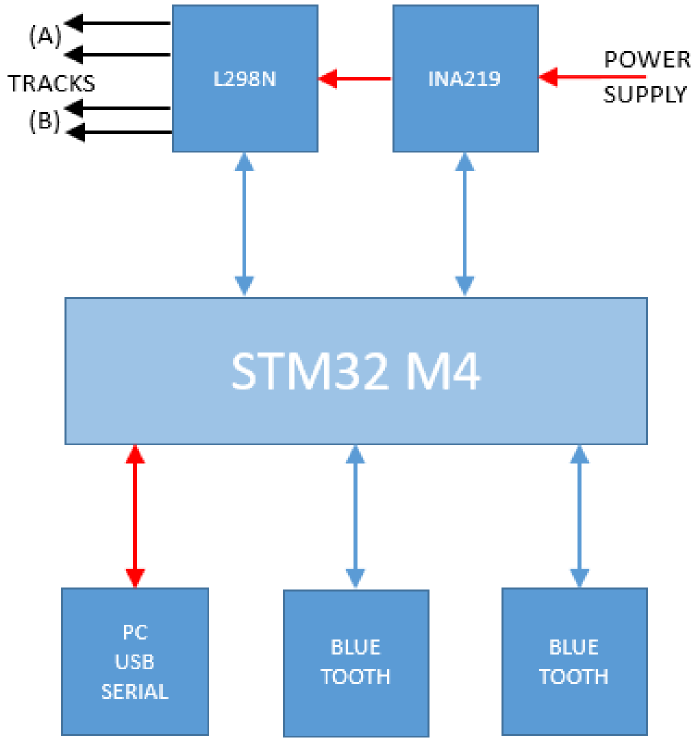
IBT-2

Example Hardware Configurations

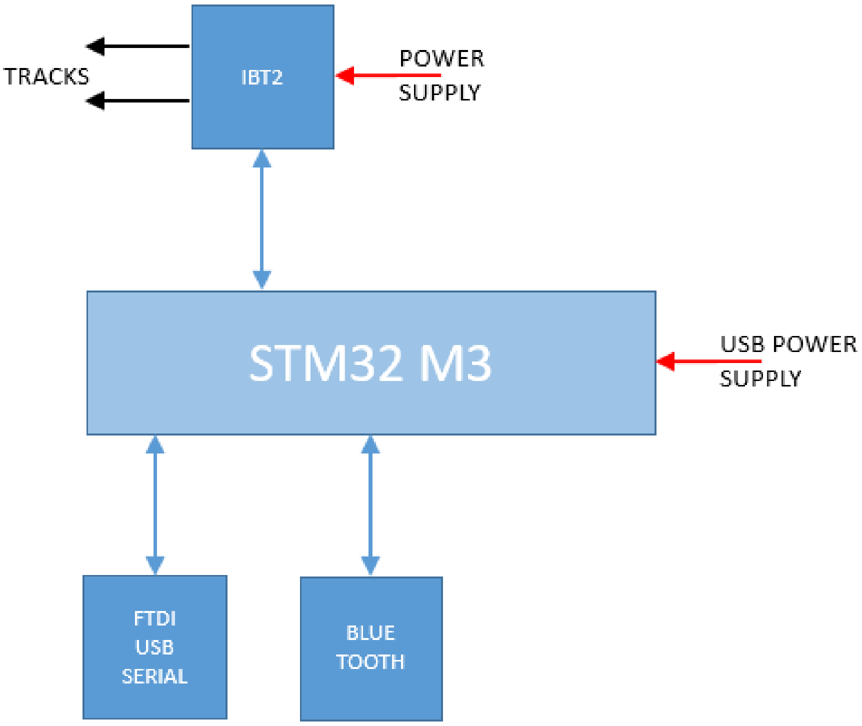
Three blue tooth connections with single 10 amp drive capability on either processor board:



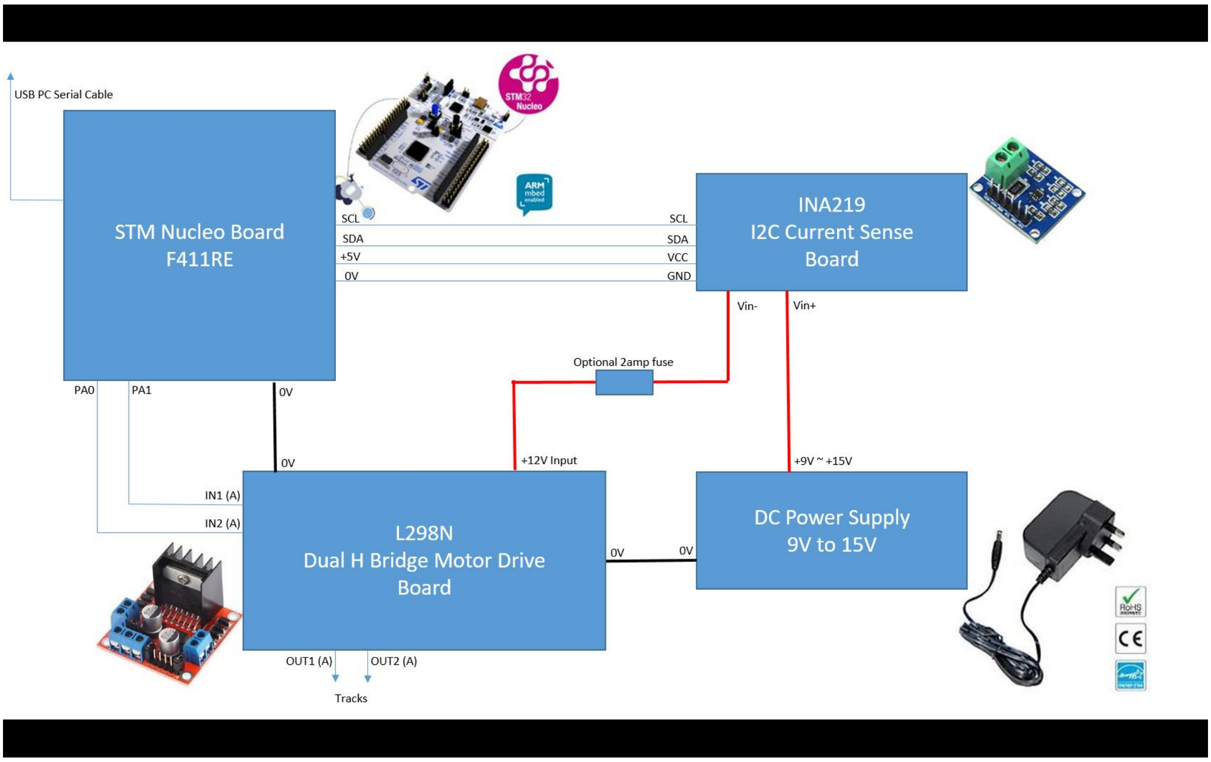
One serial PC connection and two Bluetooth connections with two track 2amp drive capability:



One serial connection, one Bluetooth connection, no service mode available (no INA219) and 10amp drive capability:



The following block diagram shows the individual components and required inter-connectivity for one output operation. To add service mode on track B output you will need to connect extra wires from the F411RE to the L298N.



We always recommend a fuse is added between the INA219 and the H Bridge to protect circuits and trains.

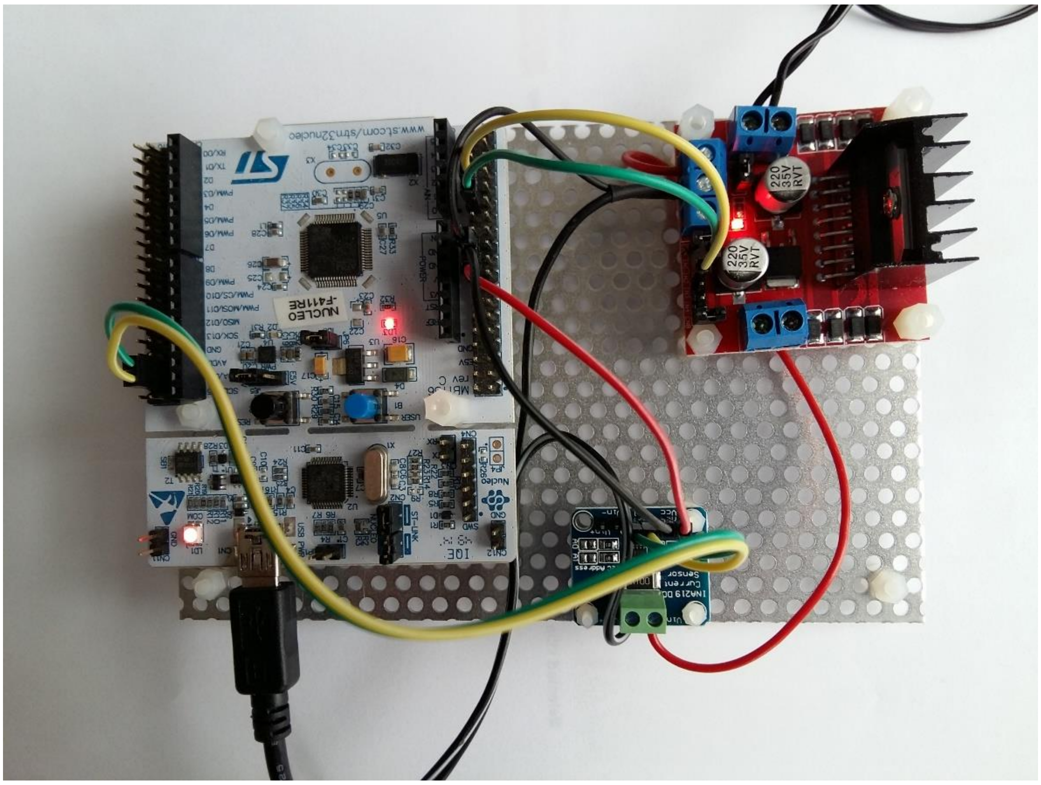
Please choose a power supply with a voltage suitable for your gauge. If in doubt consult the NMRA website for recommendations. The following table is a general guide:

Gauge	DC Supply Voltage
N	12.0V
OO/HO	14.4V
O/G	18.0V

Note however that modern engines appear to tolerate lower voltages than these.

STM32F411RE Example

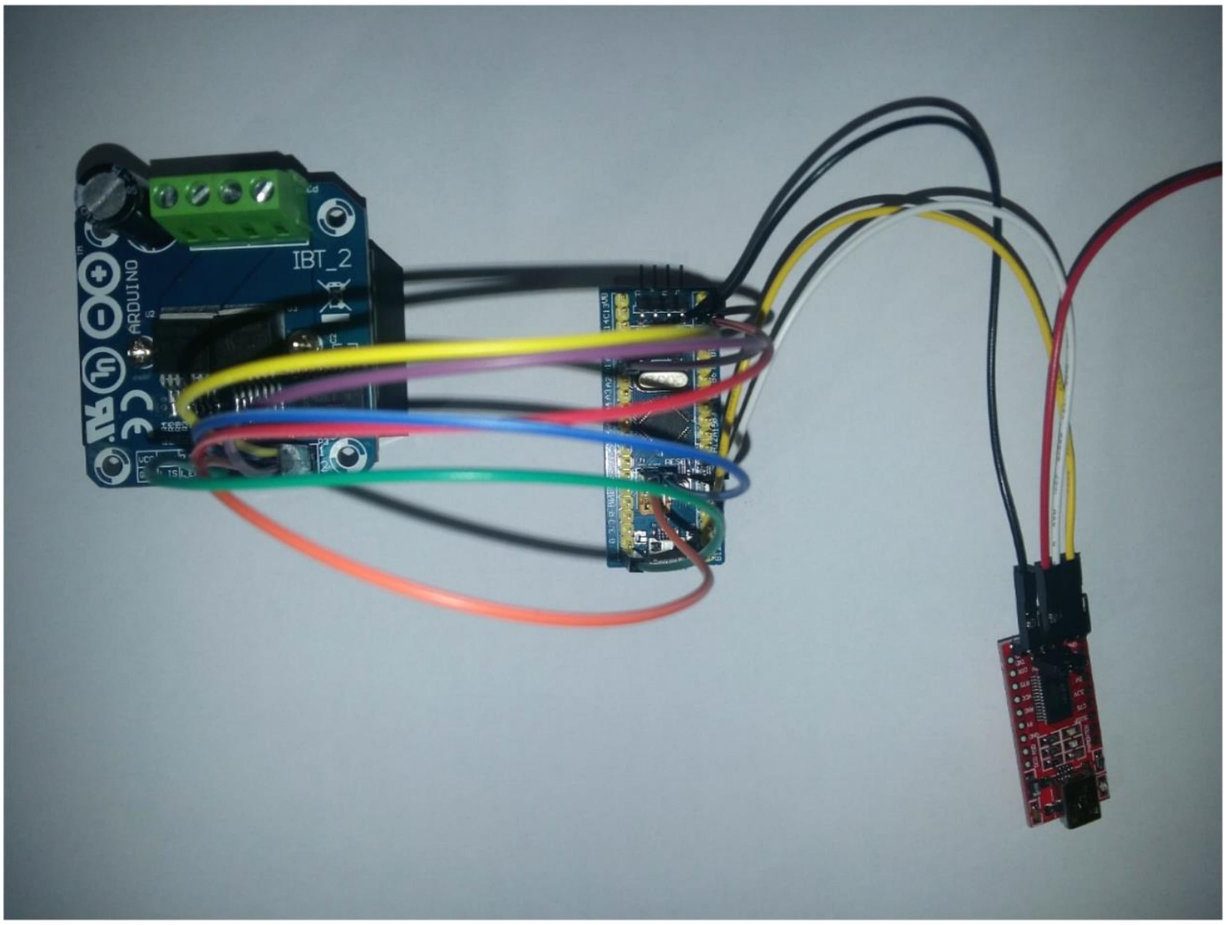
The following picture shows all components connected together to build a 2amp dual track controller with CV programming using an STM32F411RE ARM Cortex M4 board. This build requires no soldering and uses simple female to female breadboard jumpers purchased from EBay.



Please send any queries regarding building and programming to support@swws.co.uk for support.

STM32F103 Example

The following picture shows all components connected together to build a 10amp single track controller without CV programming using an STM32F103 ARM Cortex M3 board. This build requires some soldering as the STM32F103 board does not come with any connectors attached. Please note that the IBT-2 board is a 43amp capable H bridge and we have only tested our design to 10amps drive capability. We would highly recommend a 10amp fuse is placed in line with the supply to the IBT-2.



The components from left to right in the picture above are: IBT-2 H 43amp Bridge, STM32F103 Arduino Nano board and a USB to TTL FT232RL FTDI Serial Adapter Converter Module. The USB to TTL FT232RL FTDI Serial Adapter Converter Module connects to a PC and provides a serial interface to the STM32F103 board.

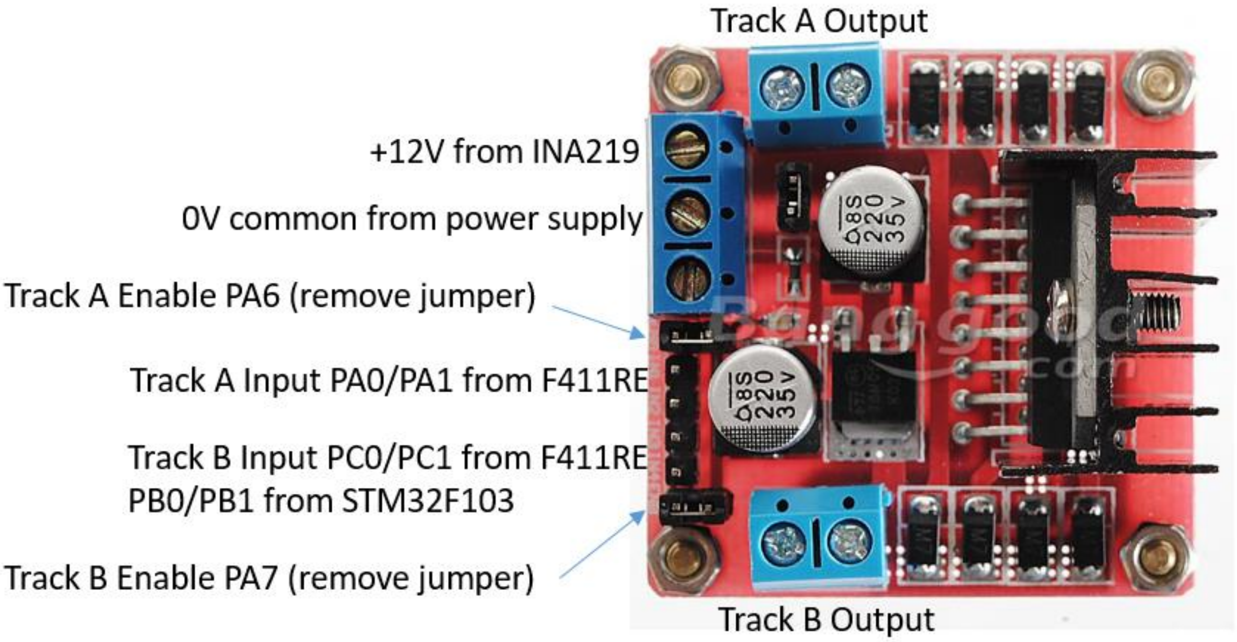
To power the SMT32F103 board you must either connect the +5volt pin from a FTDI adapter card to the 5V pin on the STM32F103 board or alternatively connect a USB power supply to the USB connector on the STM32F103 or power from the L298N +5V connector.

Please send any queries regarding building and programming to support@swws.co.uk for support.

Component Connections

This section gives details on component connections for various different components.

L298N Connections


Track A Output

+12V from INA219

0V common from power supply

Track A Enable PA6 (remove jumper)

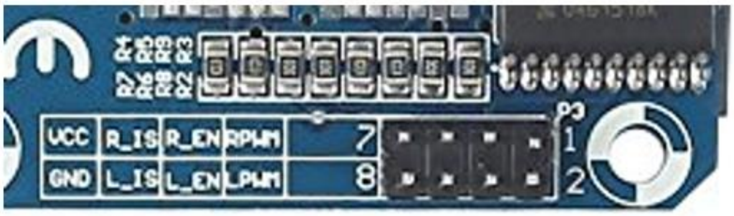
Track A Input PA0/PA1 from F411RE

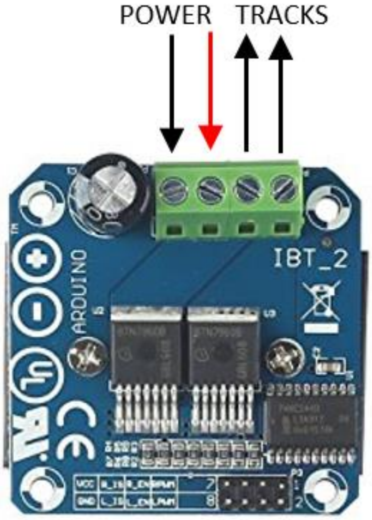
Track B Input PC0/PC1 from F411RE
PB0/PB1 from STM32F103

Track B Enable PA7 (remove jumper)

Track B Output

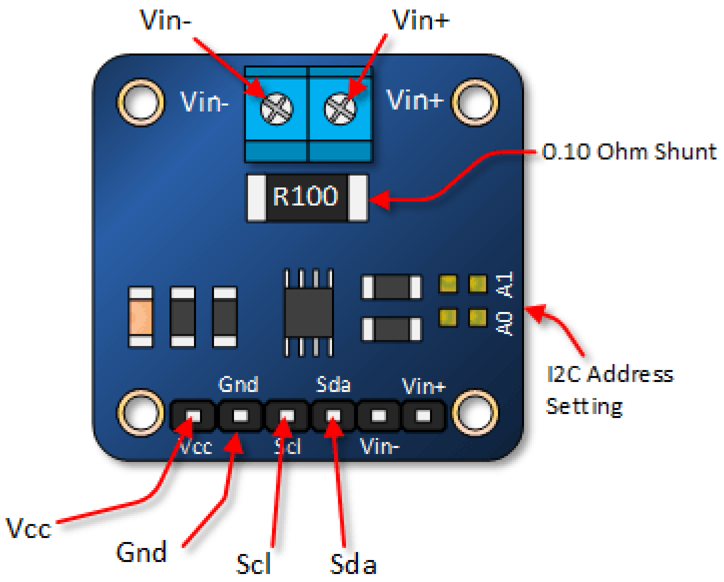
IBT-2 Connections





Pin 1 & 2 to STM32F PA0 and PA1
Pin 3 & 4 to PA6
Pin 5 & 6 not connected
Pin 7 to +5V
Pin 8 to 0V or GND

INA219 Connections



IN219 Connection	Connection To
Vin+	Power supply positive
Vin-	+12V L298N
Vcc	F411RE +3.3V CN7 Pin 16 or 3.3 pin on STM32F103
Gnd	OV common of L298N or F411RE GND Pin
Scl	F411RE SCL CN10 Pin3 or B6 on STM32F103
Sda	F411RE SDA CN10 Pin 5 or B7 on STM32F103

NOTE: The I2C address default is used in the design, so no connections are needed for A0 or A1 pads.

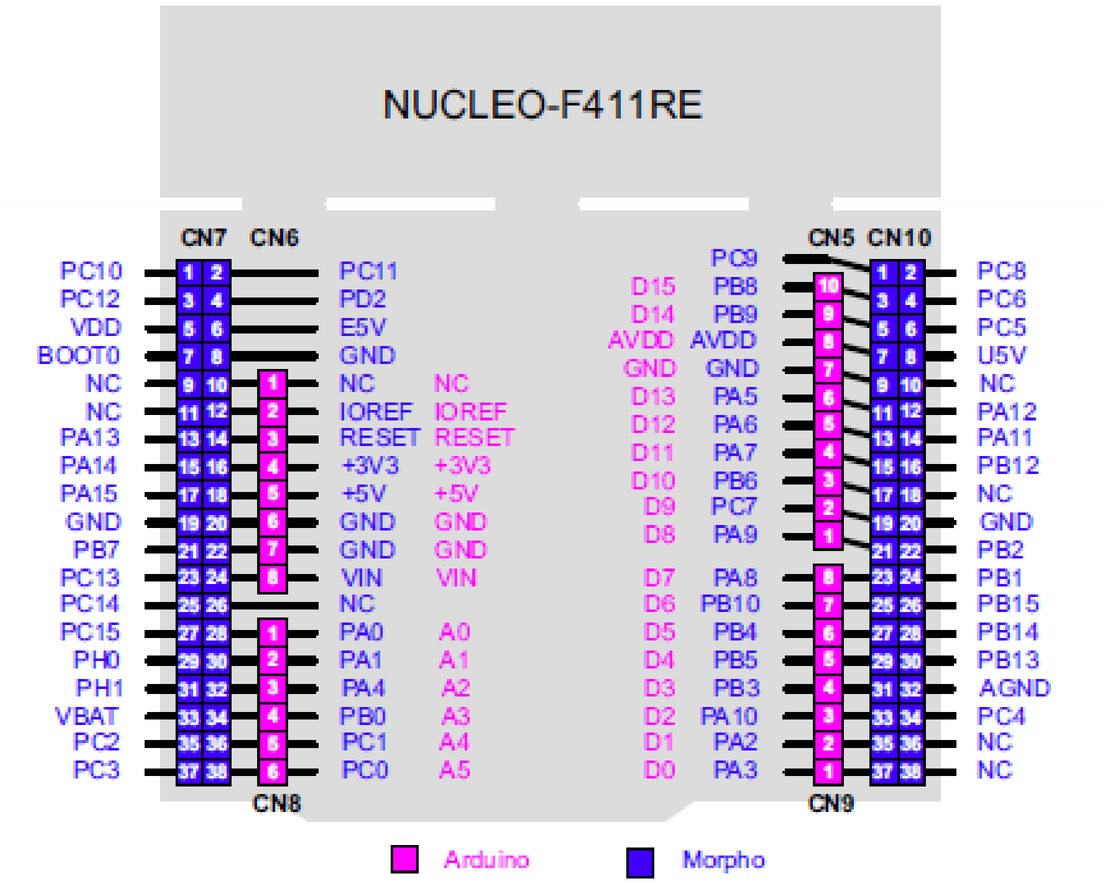
NOTE: When using the IBT-2 and INA219 at currents above 3.2Amps extra R100 Ohm shunt resistors must be added in parallel to the R100 on the board. So an extra one resistor will increase the maximum current to 6.4Amps.

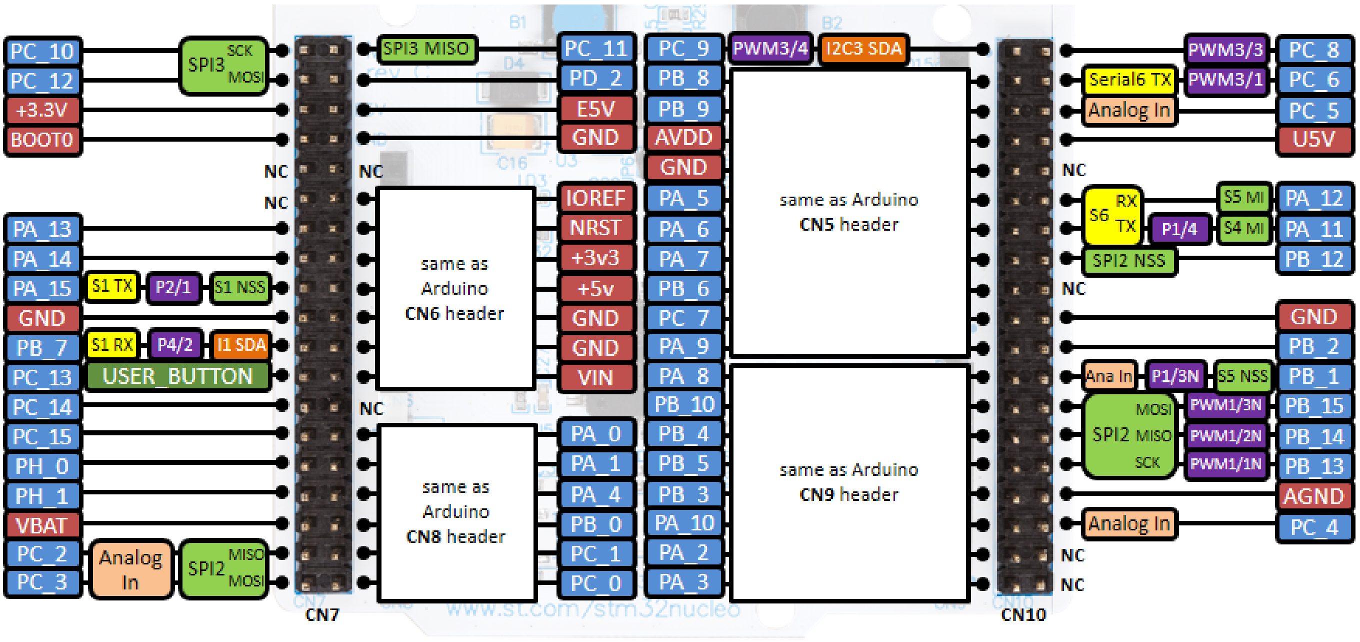
Fuse Protection

We recommend an optional inline fuse is connected to the power supply for short circuit protection. These are readily available on EBay. We recommend a 20mm 2Amp fuse as show below with inline fuse holder:

STM Nucleo F411RE Connections

The following shows all the F411RE connections, the table below defines required connections to other modules.

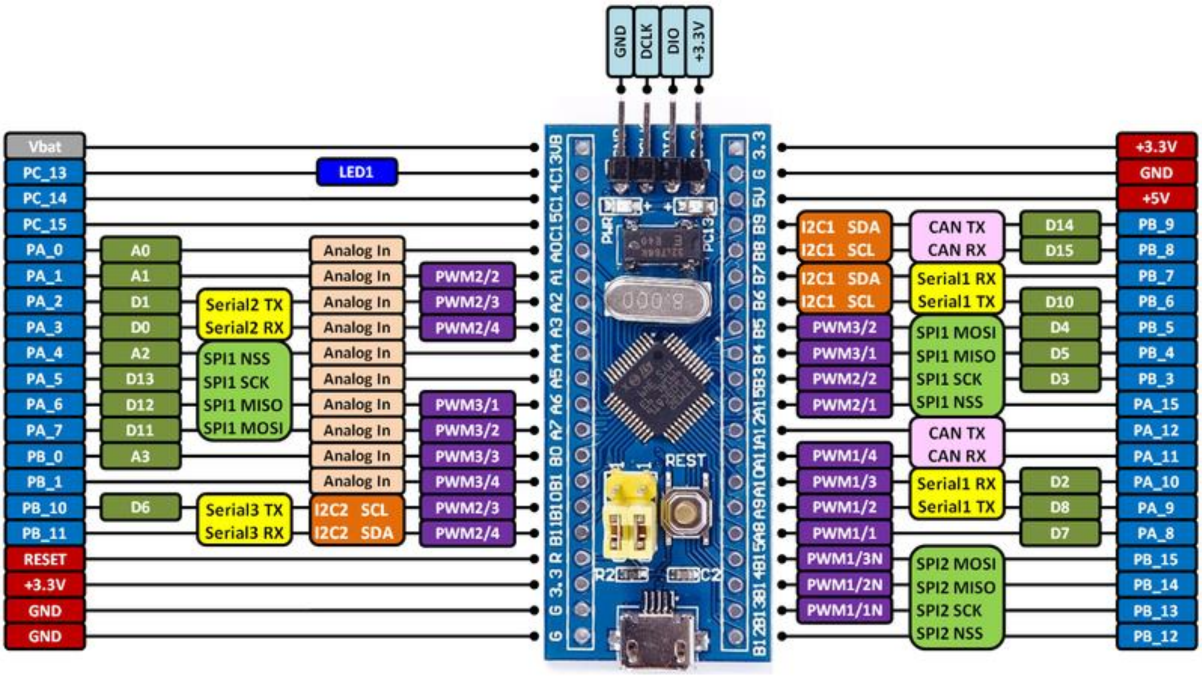




F411RE PIN	F411RE Connection	Connection To
PA0	CN7 Pin 28[3]	L298N IN1 Motor A
PA1	CN7 Pin 30[3]	L298N IN2 Motor A
PC0	CN7 Pin 38[1]	L298N IN3 Motor B
PC1	CN7 Pin 36[1]	L298N IN4 Motor B
PB8/SCL	CN10 Pin3	INA219 SCL
PB9/SDA	CN10 Pin 5	INA219 SDA
3.3V	CN7 Pin16	INA219 VCC
GND	CN10 Pin20	INA219 GND
GND	CN7 Pin8	L298N 0V
PA6	CN10 Pin13[2]	L298N ENA Enable
PA7	CN10 Pin15	L298N ENB Enable
PA11	CN10 Pin14[4]	USART6 Tx Data
PA12	CN10 Pin12[4]	USART6 Rx Data
PA9	CN10 Pin21[4]	USART1 Tx Data
PA10	CN10 Pin33[4]	USART1 Rx Data

1. Only required when using separate track output B for service mode.
2. Can be connected to IBT-2 R_EN and L_EN, used with overload detection or as simple enable outputs and track power control.
3. Can be connected to IBT-2 LPWM and RPWM.
4. Can be connected to either a Bluetooth module or a FTDI USB 232 Serial Adapter Converter Module.

STM32F103 Connections



STM32F103 PIN	Board Pin Label	Connection
PA0	A0[3]	L298N IN1 Motor A
PA1	A1[3]	L298N IN2 Motor A
PB0	B0[1]	L298N IN3 Motor B
PB1	B1[1]	L298N IN4 Motor B
PB6 SCL I2C1	B6	INA219 SCL
PB7 SDA I2C1	B7	INA219 SDA
3.3V	3.3	INA219 VCC
GND	G	INA219 GND
GND	G	L298N 0V
PA9	A9[2]	UART1 Tx Data
PA10	A10[2]	UART1 Rx Data
PB10	B10[2]	UART3 Tx Data
PB11	B11[2]	UART3 Rx Data
PA2	A2[2]	UART2 Tx Data
PA3	A3[2]	UART2 Rx Data
PA6	A6[4]	L298N ENA Enable
PA7	A7	L298N ENB Enable

1. Only required when using separate track output B for service mode.
2. Can be connected to either a Bluetooth module or a FTDI USB 232 Serial Adapter Converter Module.
3. Can be connected to IBT2 LPWM and RPWM.
4. Can be connected to IBT-2 R_EN and L_EN, used with overload detection or as simple enable outputs and track power control.

Overload Detection

The DCC control system can provide overload detection. This is achieved by monitoring the current using the INA219 and disabling the H Bridge drive if a current that is too high is detected.

To use the overload detection the SM32F411 or STM32F103 pins PA6 and PA7 must be connected to the H Bridge enable pins. So for the L298N connect PA6 to ENA and PA7 to ENB pins. For the IBT2 H Bridge connect PA6 to R_EN and L_EN.

To enable the overload current limit detection use either the Windows or Android configuration screens.

Please also use a fuse rated at or near the current limit for the INA219 or your power supply whichever is the lower. The INA219 can be configured to operate at 6.4amp, 9.6amp or 12.8amp by adding extra shunt resistors of 0.1 ohm 2W value. For 6.4amp add one extra shunt resistor, for 12.8amp add an extra three shunt resistors, for 9.6amp add an extra two resistors.

If an overload is detected in the Windows application a warning message will appear. To reset the overload detection click the **OK** button. To cancel the overload detection click the **Cancel** button.

The Android application will only warn that an overload has been detected.

NMRA DCC Compliance

Engine Address

The DCC controller system supports 7 bit multi-function decoder address range from 1 to 127.

Speed Steps

The DCC controller system supports 28 speed steps (engine decoder CV value 29 bit 1 set).

Accessory Decoder Address

The DCC controller supports 9 bit accessory decoder address range from 1 to 512.

Engine Decoder Functions

The DCC controller currently supports NMRA DCC function groups for FL (F0) and F1 to F28.

FL and F1 to F4 are implemented using NMRA DCC packet Function Group One Instruction (100) as described in NMRA standard S-9.2.1 July 2012.

F5 to F12 are implemented using NMRA DCC packet Function Group Two Instruction (101) as described in NMRA standard S-9.2.1 July 2012.

F13 to F20 are implemented using NMRA DCC packet Feature Expansion Instruction (110) as described in NMRA standard S-9.2.1 July 2012.

F21 to F28 are implemented using NMRA DCC packet Feature Expansion Instruction (110) as described in NMRA standard S-9.2.1 July 2012.

Serial Interface Command Reference

Single Character Commands:

Command Character	Comment/Function
1..9	Set engine address from one to nine, above nine use EA command
+	Increase engine speed step by one
-	Decrease engine speed step by one
<	Program selected engine address reverse
>	Program selected engine address forward
X	Generate DCC reset packet
!	Emergency stop for all engines
0	Stop engine for selected engine address

Multi Character Commands:

Command String	Comment/Function
SM	Enter service mode using L298N track output A
SMB	Enter service mode using L298N track output B
OP	Enter operational mode (default mode) using L298N track output A
IDL	Return to idle from either operational or service mode
AC	Accessory command followed by address and value, e.g. AC12=1
RV	Read CV value followed by CV address, e.g. RV1
WV	Write CV value followed by address and value, e.g. WV1=3
EA	Set engine address for following commands, e.g. EA21 or EA6?
S	Set current addressed engine speed, e.g. S20, can use S0 for stop
F	Function on/off command, e.g. F0=1 (on) F1=0 (off), FL is same as F0
FA	Function group 1 command followed by value, e.g. FA=16
FB	Function group 2 command followed by value, e.g. FB=1 (F5), FB=128 (F12)
FC	Function group F13..F20 command followed by value, e.g. FC=16
FD	Function group F21..F28 command followed by value, e.g. FD=16
FR	Function group command repeat setting, e.g. FR=3
AR	Accessory repeat setting, e.g. AR=2
?VER	Show DCC controller version string
?AMP	Show current, pre ACK and maximum ACK readings from INA219
?APO	Show current ACK Pulse Offset setting
APO	Set ACK Pulse Offset threshold detection, e.g. APO=400
D16	Display 16 CV values from current CV address
DALL	Display all CV values
PWRA	Turn track A power on (PWRA=1) or off (PWRA=0)
PWRB	Turn track B power on (PWRB=1) or off (PWRB=0)
OLP	Set overload detection current limit value, e.g. OLP=10000 (about 1amp)
OLRST	Overload reset command
ENLK	Engine locks on/off, e.g. ENLK=1 (locks on), ENLK=0 (locks off)
ACLK	Accessory locks on/off, e.g. ACLK=1 (locks on), ACLK=0 (locks off)

All multi character commands must be terminated by a line feed, single character commands do not need any termination.

Engine address commands will cause the DCC controller to return the engine status for the address as follows: ES:<Address><Speed and Direction Byte>:<Function FA Bits>:<Function FB Bits>:<Function FC Bits>:<Function FD Bits>><Line Feed> for example ES:3:64:16:0:0:0.

The FA command controls function bits F1 to F4 and FL as described in the NMRA document s-9.2.1_2012_07.pdf. FA is command bit 0, F2 is command bit 1, FL is command bit 4.

0	0	0	FL	F4	F3	F2	F1

The FB command controls function bits F5 to F12 as described in the NMRA document s-9.2.1_2012_07.pdf. F5 is command bit 0, F12 is command bit 7.

F12	F11	F10	F9	F8	F7	F6	F5

The FC command controls function bits F13 to F20 as described in the NMRA document s-9.2.1_2012_07.pdf. F13 is command bit 0, F20 is command bit 7.

F20	F19	F18	F17	F16	F15	F14	F13

The FD command controls function bits F21 to F28 as described in the NMRA document s-9.2.1_2012_07.pdf. F21 is command bit 0, F28 is command bit 7.

F28	F27	F26	F25	F24	F23	F22	F21

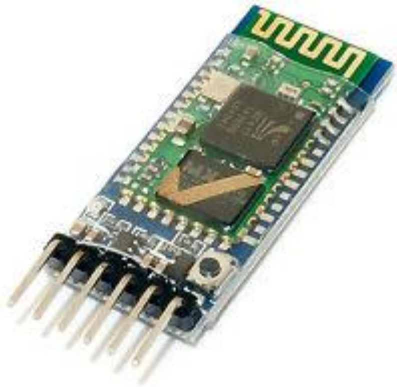
The FR (function repeat) command defines the number of times the function group 1 or function group 2 command will be transmitted onto the tracks. A value of 255 means it will be continuous. The group1 and group 2 commands are interleaved with the engine speed commands. The default value for this setting is 4. This command is provided to overcome noise and packet errors by sending multiple commands.

The AR (accessor repeat) command controls the number of times an accessory command is sent onto the tracks. The default setting for this command is 3, the valid range is 1 to 8. This command is not interleaved with engine commands. This command is provided to overcome noise and packet errors by sending multiple commands.

Bluetooth Configuration

This section describes the extra component and required configuration to operate the DCC controller via Bluetooth.

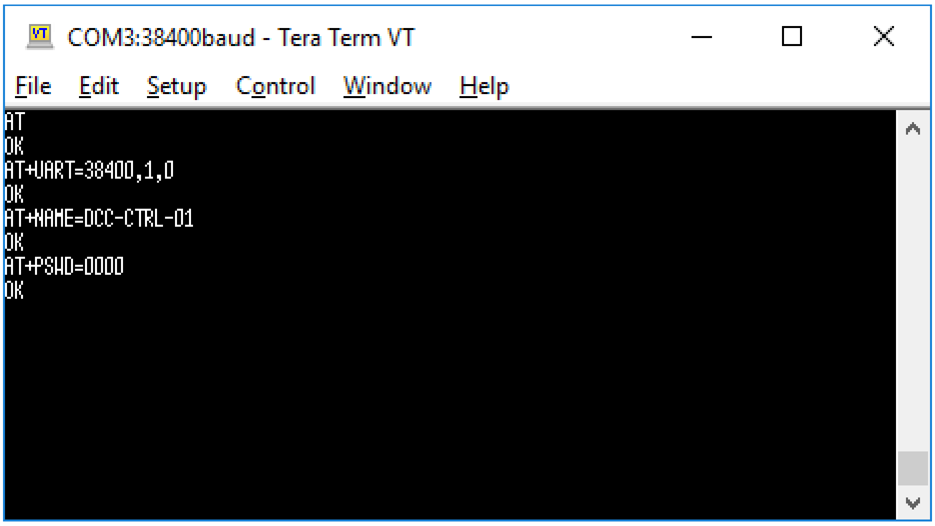
A Bluetooth module such as this is required (HC-05):



The module must be configured to operate at 38400 baud, one stop bit and no parity. This configuration is achieved by connecting the STM F411RE to a PC via the USB cable, connecting the Bluetooth module as described in the following table, loading the STM F411RE with the binary file bt_config.bin and running tera term on the PC.

The Bluetooth module must be configured in AT mode. To put the Bluetooth module into AT mode, hold down the button near the connector before turning on the STM F411RE nucleo board. The red LED on the Bluetooth module should flash once every two seconds in AT mode.

Enter the following on the tera terminal, note the name and password (pin code PSWD) can be changed from the example. If there is no response to the **AT** command then the Bluetooth module is not in AT mode.



```
AT
OK
AT+UART=38400,1,0
OK
AT+NAME=DCC-CTRL-01
OK
AT+PSWD=0000
OK
```

The FTDI USB 232 Serial Adapter module can also be used to configure the Bluetooth module. Set the FTDI VCC output to +5V and connect this to the Bluetooth power pin. Connect the GND pins together, then connect RX (FTDI) to TXD (HC-05) and TX (FTDI) to RXD (HC-05). Follow the same procedure as when using the STN32F411RE board.

UART Serial Ports

As stated previously the DCC controller boards can support up to three users via either Bluetooth or serial interfaces.

The following serial interface connections are available on the STM32F411RE DCC controller.

STM F411RE Pin	UART Function	Comment
PA11	USART6 TXD	CN10 pin 14 on nucleo board
PA12	USART6 RXD	CN10 pin 12 on nucleo board
PA9	USART1 TXD	CN10 pin 12 on nucleo board
PA10	USART1 RXD	CN10 pin 12 on nucleo board
PA2	USART2 TXD	CN10 pin 12 on nucleo board (uses USB interface)
PA3	USART2 RXD	CN10 pin 12 on nucleo board (uses USB interface)

The following connections must be made between a Bluetooth module and the STM411RE nucelo board for USART6 operation:

STM F411RE Pin	UART Function	Comment
PA11	USART6 TXD	Bluetooth RX data, CN10 pin 14 on nucleo board
PA12	USART6 RXD	Bluetooth TX data, CN10 pin 12 on nucleo board
GND	GND	0volts, CN6 pin 6
+5V	VCC	+5volts, CN6 pin 5

for USART1 operation:

STM F411RE Pin	UART Function	Comment
PA9	USART1 TXD	Bluetooth RX data, CN10 pin 14 on nucleo board
PA10	USART1 RXD	Bluetooth TX data, CN10 pin 12 on nucleo board
GND	GND	0volts, CN6 pin 6
+5V	VCC	+5volts, CN6 pin 5

The following serial connections are available on the STM32F103 board:

STM32F103 Pin	Comment
A9	Tx Data
A10	Rx Data
B10	Tx Data
B11	Rx Data
A2	Tx Data
A3	Rx Data

If using an STM32F411RE board either load dcc_ctrl_bt.bin to allow BlueTooth module configuration or use and FTDI 232 adapter module. If using an STM32F103 board then use a FTDI 233 adapter to configure the BlueTooth.

Once the STM F411RE has been programmed with the dcc_ctrl_bt.bin application the Bluetooth module can be configured using a Windows terminal emulator like teraterm.

When running the Windows application the serial port associated with the Bluetooth module must be selected for communications.

There is no need for a PC connection when using only BlueTooth serial connections unless the USB cable is being used to power the STM32F411RE board.

STM32F103 & STM32F411RE Board LEDs

The board LEDs are used to indicate status. When in operational mode or idle mode the LEDs flash once every two seconds to indicate the software is running correctly. In service mode the board LEDs only flash during CV reading to indicate an acknowledge pulse is being read from the decoder. If there is a current overload detected the software will disable the H Bridge and flash the LEDs four times a second.

Multi User Operation

The DCC controller can support up to three users at any one time. One connected via the serial interface and two connected via Bluetooth interfaces. On the STM32F103 there can be three BlueTooth connections only.

The users cannot share engines and only one user can change the DCC controller state from idle to operational or service mode. Only one user can perform service mode operations.

The user that is first to select either operational or service mode is considered the state owner and only he or she can change the state of the DCC controller.

Once a user has selected an engine address and sent any command (speed/stop/function) he or she owns that address until the DCC controller returns to the idle state.

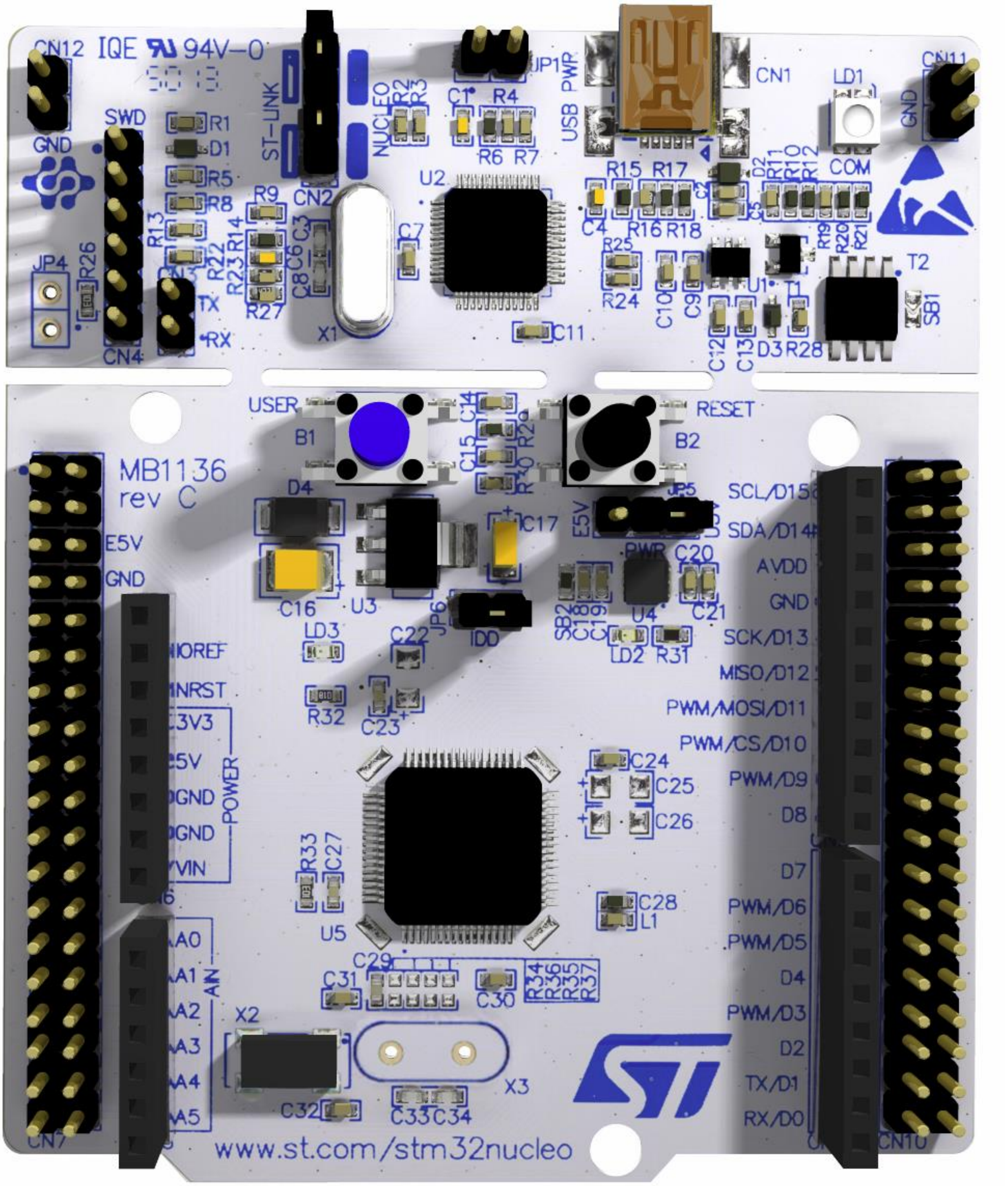
If a user attempts to control an already owned engine the Windows or Android application will indicate the engine address is locked.

Engine locking can be turned on or off using the Windows or Android application configuration window. The same is true for accessory locks. These options allow one user multiple connection to the DCC controller so walk about mode can be used.

Power STM32F411RE from L298N

When operating the Bluetooth version of this design the need for a USB cable between a PC and the STM32F411RE board can be removed by using the +5V output on the L298N board.

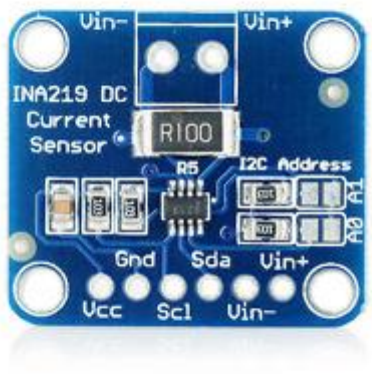
Move the jumper located near the RESET button (black in image below) from U5V to E5V. Connect the L298N +5V output to the STM32F411RE board E5V pin (top left of main board below). If in doubt consult the STM Nucleo-64 board user manual UM1724.



IBT-2 10Amp H Bridge Option

The high power option (Arduino IBT-2) adds support for up to 10amp drive capability. The H Bridge shown below is an IBT-2 from EBay:

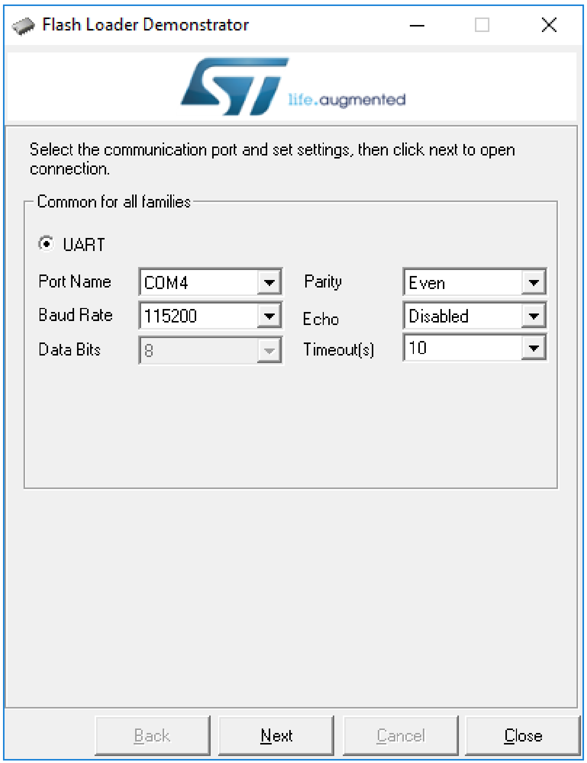
This H Bridge can actually support up to 43 amps but has only been tested by us up to 10 amps. This H Bridge replaces the L298N in the 2 amp design. In addition to replacing the H Bridge, the INA219 must be modified to handle the extra current if the INA219 is left in the design.



To allow the INA219 to handle 10 amps, three 100 milli-ohm resistors must be added in parallel to the existing R100 ohm resistor as shown above. These extra resistors can either be soldered to the board or attached between the Vin- and Vin+ connections.

Programming SMT32F103 Arduino Boards

To program STM32F103 boards you must download from the STM website the **Flash Loader Demonstrator** Windows application. The programming screens appear as follows:

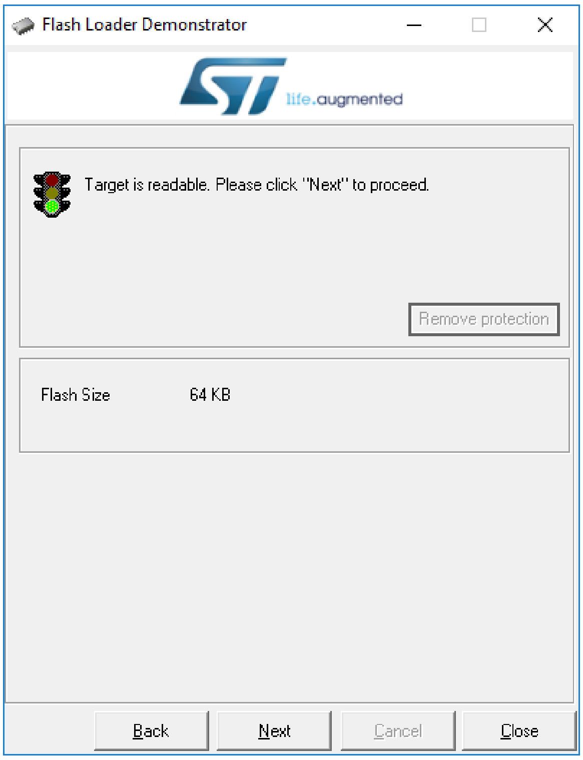


Select the communication port and set settings, then click next to open connection.

Common for all families

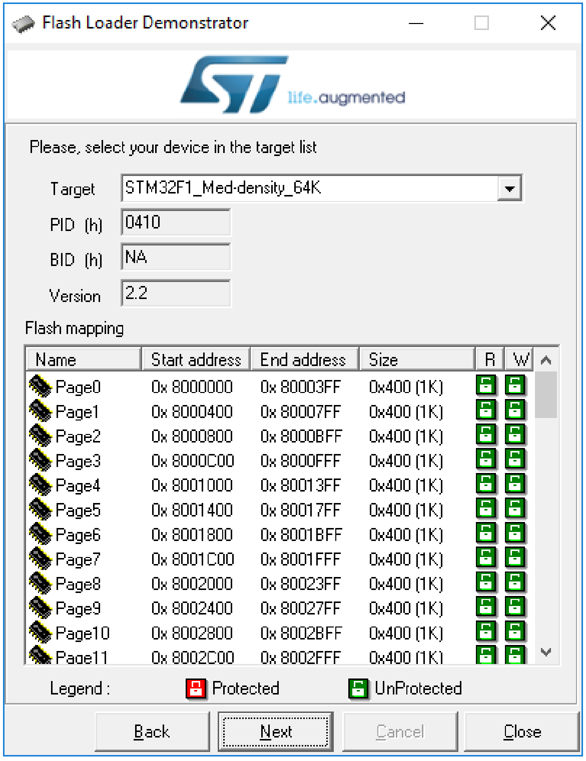
● UART

Port Name	COM4	Parity	Even
Baud Rate	115200	Echo	Disabled
Data Bits	8	Timeout(s)	10



Target is readable. Please click "Next" to proceed.

Flash Size 64 KB



Please, select your device in the target list

Target: STM32F1_Med-density_64K

PID (h): 0410

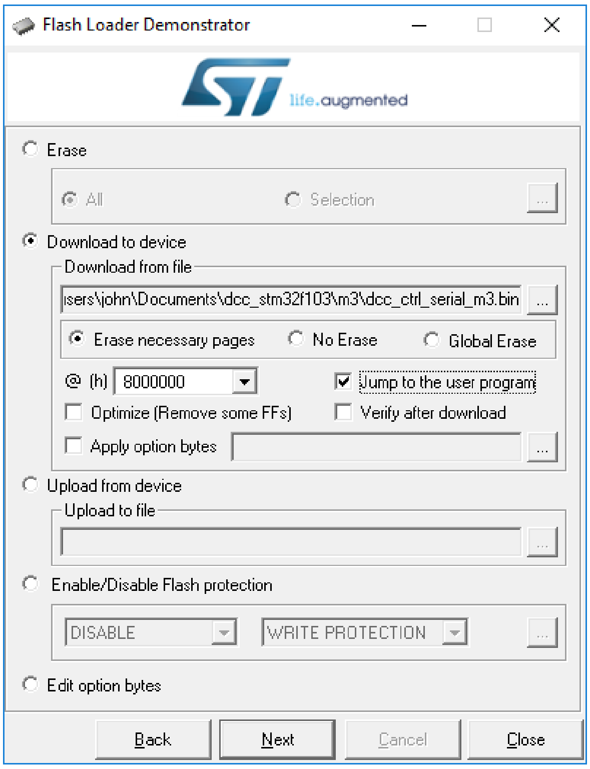
BID (h): NA

Version: 2.2

Flash mapping

Name	Start address	End address	Size	R	W
Page0	0x 8000000	0x 80003FF	0x400 (1K)	🟩	🟩
Page1	0x 8000400	0x 80007FF	0x400 (1K)	🟩	🟩
Page2	0x 8000800	0x 8000BFF	0x400 (1K)	🟩	🟩
Page3	0x 8000C00	0x 8000FFF	0x400 (1K)	🟩	🟩
Page4	0x 8001000	0x 80013FF	0x400 (1K)	🟩	🟩
Page5	0x 8001400	0x 80017FF	0x400 (1K)	🟩	🟩
Page6	0x 8001800	0x 8001BFF	0x400 (1K)	🟩	🟩
Page7	0x 8001C00	0x 8001FFF	0x400 (1K)	🟩	🟩
Page8	0x 8002000	0x 80023FF	0x400 (1K)	🟩	🟩
Page9	0x 8002400	0x 80027FF	0x400 (1K)	🟩	🟩
Page10	0x 8002800	0x 8002BFF	0x400 (1K)	🟩	🟩
Page11	0x 8002C00	0x 8002FFF	0x400 (1K)	🟩	🟩

Legend : 🟥 Protected 🟩 UnProtected



○ Erase

● All ○ Selection

● Download to device

Download from file: isers\john\Documents\dcc_stm32f103\m3\dcc_ctrl_serial_m3.bin

● Erase necessary pages ○ No Erase ○ Global Erase

@ (h) 8000000 ☑ Jump to the user program

☐ Optimize (Remove some FFs) ☐ Verify after download

☐ Apply option bytes

○ Upload from device

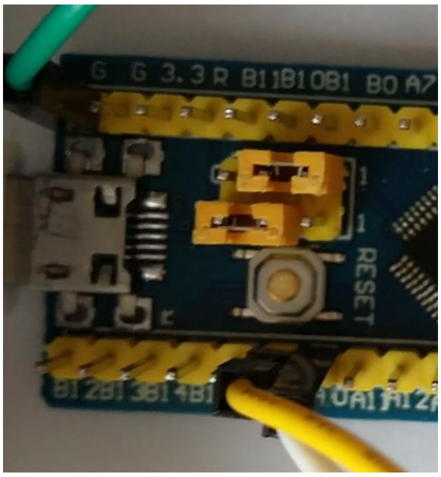
Upload to file

○ Enable/Disable Flash protection

DISABLE WRITE PROTECTION

○ Edit option bytes

To program the STM32F103 make sure the two boot jumpers near the USB connector are set as shown below:



Connect a FTDI 232 USB serial adapter as shown above on the right to A9 and A10 pins on the STM32F103. Power the STM32F103 from either the FTDI module or from a USB cable connected between a PC and the STM32F103 USB connector. **Note if using the FTDI to power the STM32F103 ensure the correct voltage (jumper on FTDI) is selected and connected to the correct pin on the STM32F103.**

Run the flash loader software on the PC, select the correct COM port for the FTDI USB module. Select the next button three times assuming all is ok (as per the previous page screens).

Select **Download to device** and open from the .zip file the binary file **dcc_ctrl_serial_m3.bin** then select the next button, a progress screen should appear. Once the programming is complete change the boot jumpers so both jumpers are near the USB connector.

Cycle power on the STM32F103 board and then follow the programming verification section.

Programming STM32F411RE Nucleo Boards

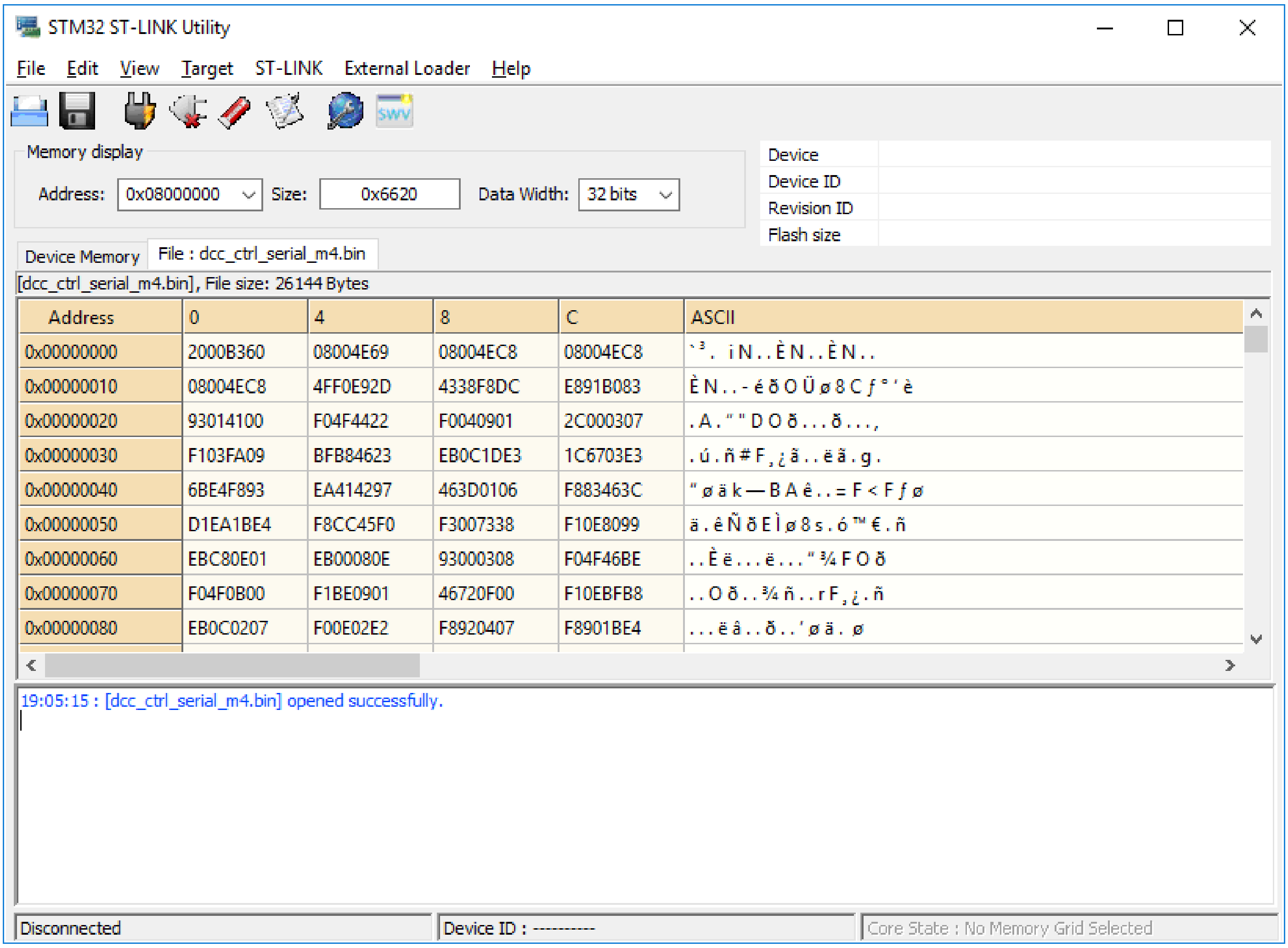
To program STM32F411RE boards you must download from the STM website the **STM32 ST-Link Utility** Windows application and associated USB drivers.

The board is then programed using the STM32 ST-LINK Utility as follows:

Install the STM V2 Link software, this can be downloaded for free from: http://www.st.com/en/embedded-software/stsw-link004.html.

Connect the F411RE board to a PC using a USB cable.

Use the file menu and open the binary file you require to load, this is **dcc_ctrl_serial_m4.bin** for the DCC controller or **bt_config.bin** if you need to configure a Bluetooth module.



Address: 0x08000000 Size: 0x6620 Data Width: 32 bits

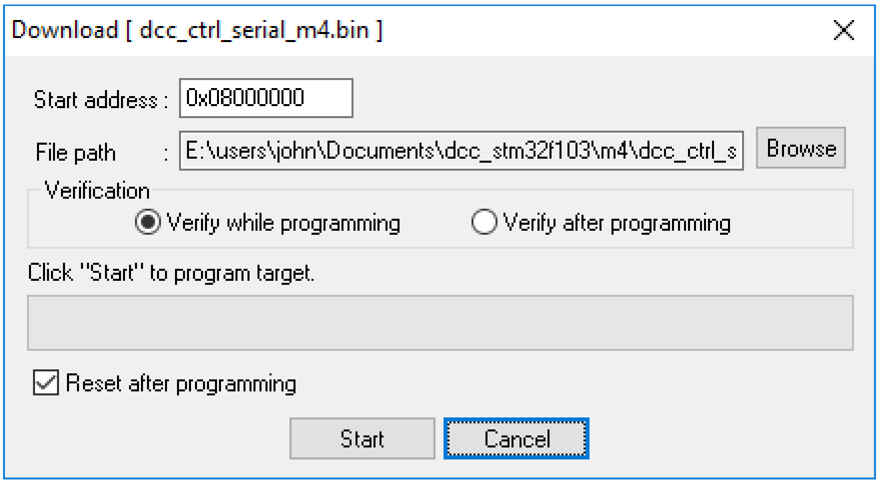
[dcc_ctrl_serial_m4.bin], File size: 26144 Bytes

Address	0	4	8	C	ASCII
0x00000000	2000B360	08004E69	08004EC8	08004EC8	`³. iN..ÈN..ÈN..
0x00000010	08004EC8	4FF0E92D	4338F8DC	E891B083	ÈN..-éðOÜø8Cƒ°'è
0x00000020	93014100	F04F4422	F0040901	2C000307	.A."" DOð...ð...,
0x00000030	F103FA09	BFB84623	EB0C1DE3	1C6703E3	.ú.ñ#F¸¿ã..ëã.g.
0x00000040	6BE4F893	EA414297	463D0106	F883463C	"øäk—BAê..=F<Fƒø
0x00000050	D1EA1BE4	F8CC45F0	F3007338	F10E8099	ä.êÑðEÌø8s.ó™€.ñ
0x00000060	EBC80E01	EB00080E	93000308	F04F46BE	..Èë...ë..."¾FOð
0x00000070	F04F0B00	F1BE0901	46720F00	F10EBFB8	..Oð..¾ñ..rF¸¿.ñ
0x00000080	EB0C0207	F00E02E2	F8920407	F8901BE4	...ëâ..ð..'øä. ø

19:05:15 : [dcc_ctrl_serial_m4.bin] opened successfully.

Disconnected | Device ID : ---------- | Core State : No Memory Grid Selected

Use the target menu then program and verify (CTRL+P) to program to F411RE flash memory:



Download [dcc_ctrl_serial_m4.bin]

Start address : 0x08000000

File path : E:\users\john\Documents\dcc_stm32f103\m4\dcc_ctrl_s

Verification

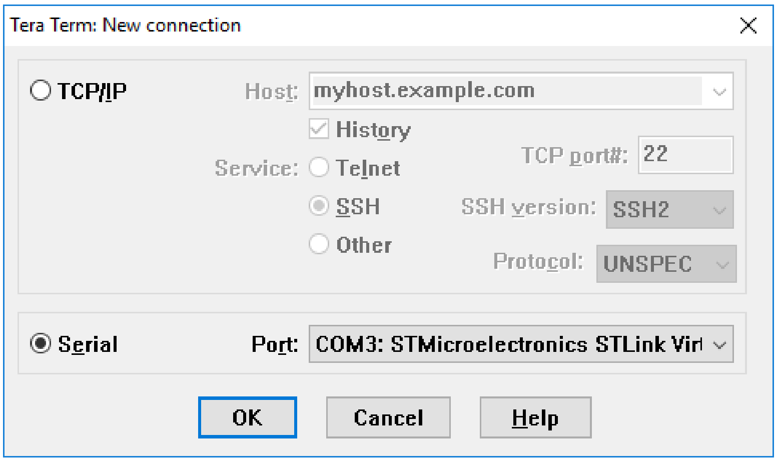
● Verify while programming ○ Verify after programming

Click "Start" to program target.

☑ Reset after programming

Programming Verification

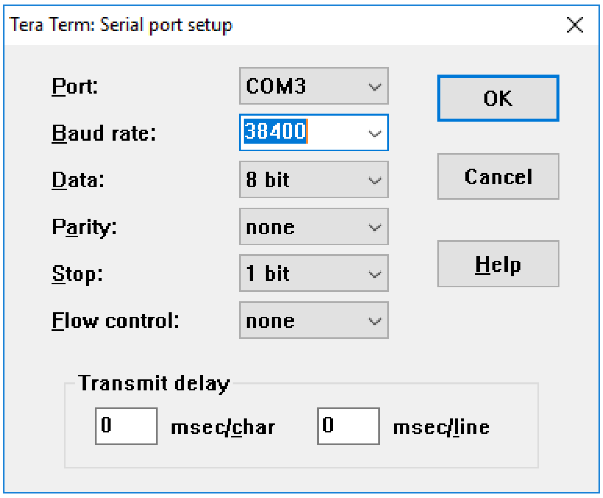
Once the board has been programmed and if required the boot jumpers changed (STM32F103) the board LED should flash once a second. To further verify the board has programmed use Tera Term or any other terminal emulator program and connect to the board serial port either using a USB cable for the STM32F411RE or an FTDI 232 module for the STM32F103 board. Set Tera Term up as follows:



Tera Term: New connection

○ TCP/IP   Host: myhost.example.com
☑ History
Service: ○ Telnet   TCP port#: 22
● SSH   SSH version: SSH2
○ Other   Protocol: UNSPEC

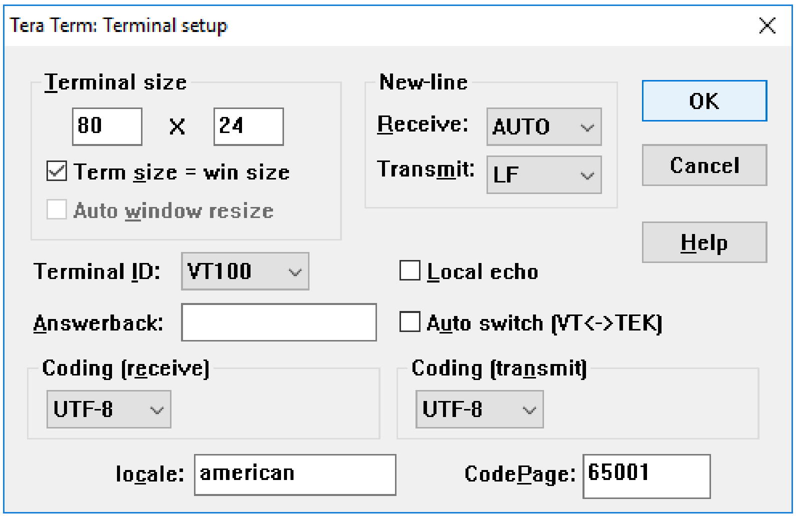
● Serial   Port: COM3: STMicroelectronics STLink Virt



Tera Term: Serial port setup

Port: COM3
Baud rate: 38400
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char   0 msec/line



Tera Term: Terminal setup

Terminal size
80 X 24
☑ Term size = win size
☐ Auto window resize

New-line
Receive: AUTO
Transmit: LF

Terminal ID: VT100   ☐ Local echo
Answerback: ☐ Auto switch (VT<->TEK)

Coding (receive)   Coding (transmit)
UTF-8   UTF-8

locale: american   CodePage: 65001

To verify correct operation of the software use the following instructions on the terminal:

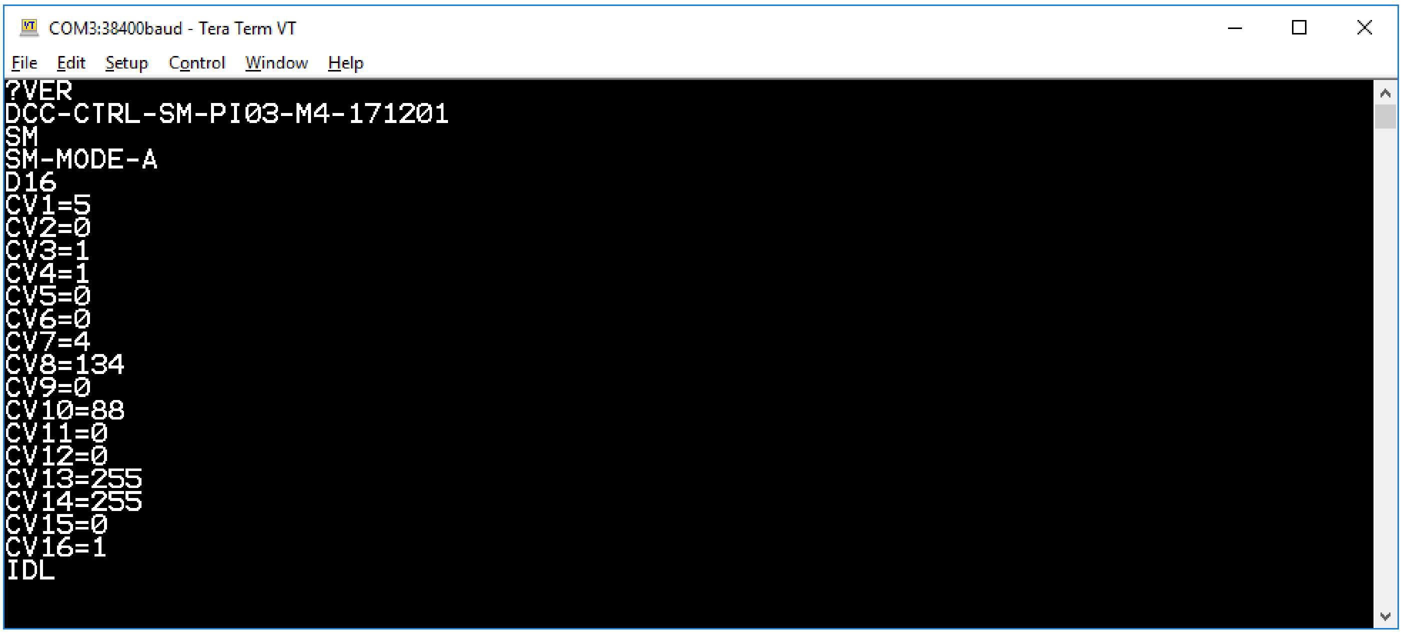
Enter **?VER** to display the software version number

Enter **SM** to enter service mode A

Enter **D16** to dump the first 16 CV registers

Enter **IDL** to return to idle mode

The output should appear as follows on the Tera Term display:



COM3:38400baud - Tera Term VT
File Edit Setup Control Window Help

```
?VER
DCC-CTRL-SM-PI03-M4-171201
SM
SM-MODE-A
D16
CV1=5
CV2=0
CV3=1
CV4=1
CV5=0
CV6=0
CV7=4
CV8=134
CV9=0
CV10=88
CV11=0
CV12=0
CV13=255
CV14=255
CV15=0
CV16=1
IDL
```

Software Applications

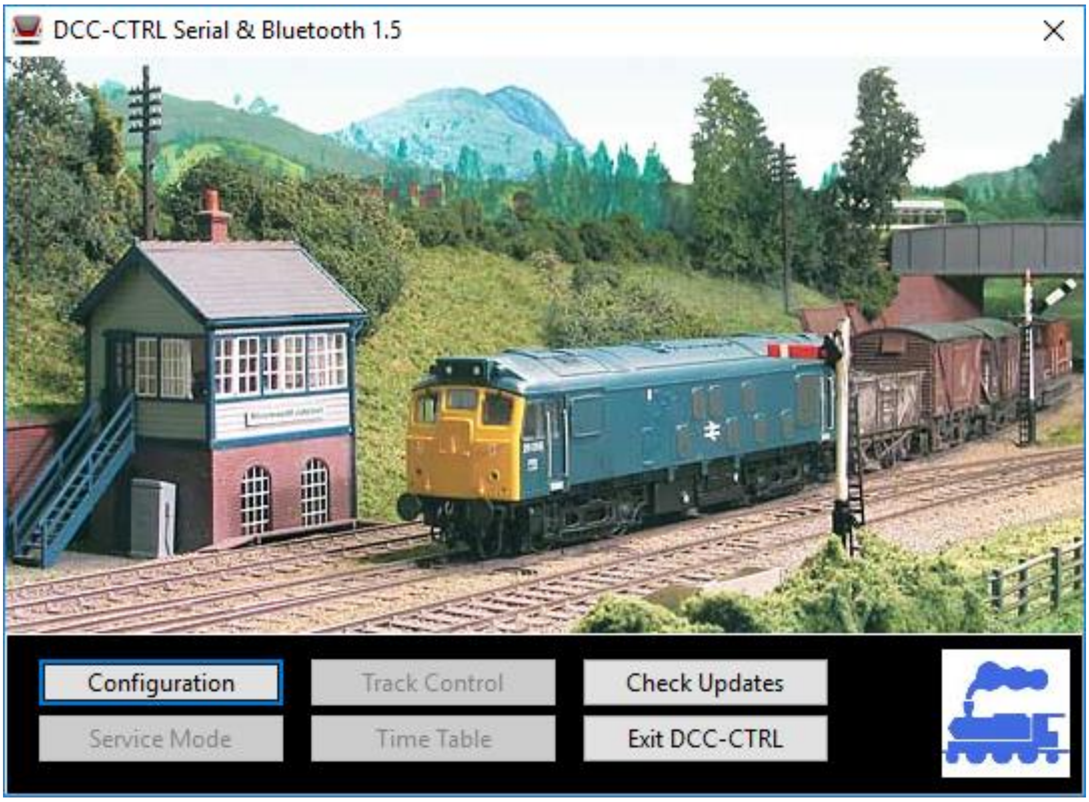
Windows Application

The Windows application is stored in the .zip file as dcc_ctrl.exe. This application does not need to be installed it can be copied from the .zip and placed on any of the PC drives. The Windows application allows for CV programming (service mode), train/points control and a time table driven mode for trains and points.

The following sections explain the different screens.

Start Screen

When the Windows application is executed the screen appears as follows:



This screen provides the user with buttons to configure the DCC control system, check for software updates on the internet, exit the application and run the different operating modes of the DCC control software.

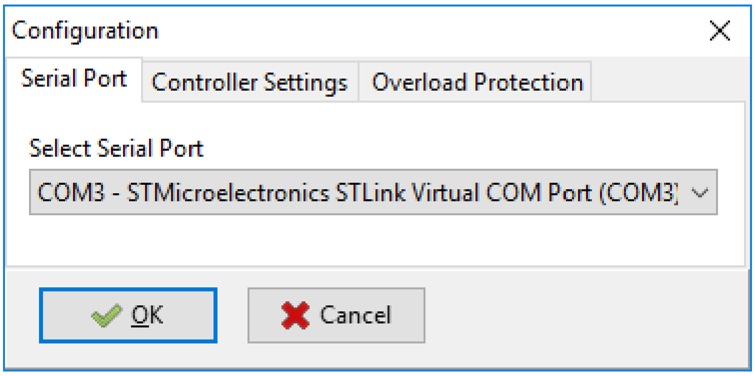
The check updates button will interrogate the www.swws.co.uk website to establish if there is a newer version of the software available. If there is a newer version the user will be informed of the download URL link and the new version number and release date.

The exit button closes the Windows application.

The configuration button displays a configuration window which is explained in the next section.

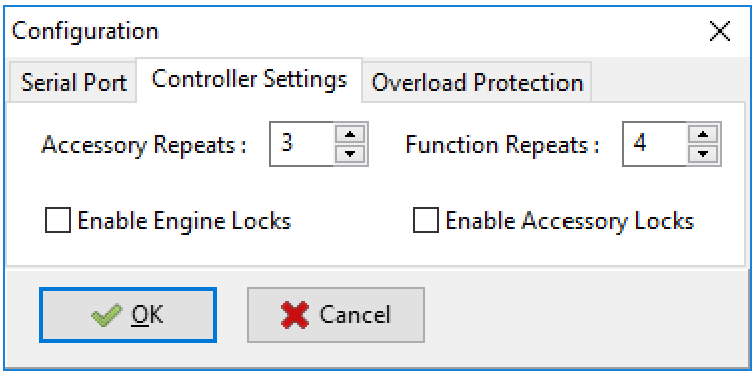
Configuration Window

The configuration screen has two pages. The first place allows the user to choose a serial port that is used to communicate commands to the DCC control system:



Select Serial Port: COM3 - STMicroelectronics STLink Virtual COM Port (COM3)

The second page allows for a number of DCC control configuration options:



Accessory Repeats : 3

Function Repeats : 4

□ Enable Engine Locks

□ Enable Accessory Locks

The four options available are explained below:

Accessory Repeats

As DCC is unreliable due mainly to the mechanical pickup on the track and dirt on the track the user can configure the number of times an accessory packet is sent to the decoder. The maximum number of accessory packets that can be sent is 8. The default value is shown above.

Function Repeats

This controls the number of times a function packet is sent to a train decoder. A function repeat value of 255 means that the function command is sent all the time (after the engine speed packet) to the decoder. The default value is shown above.

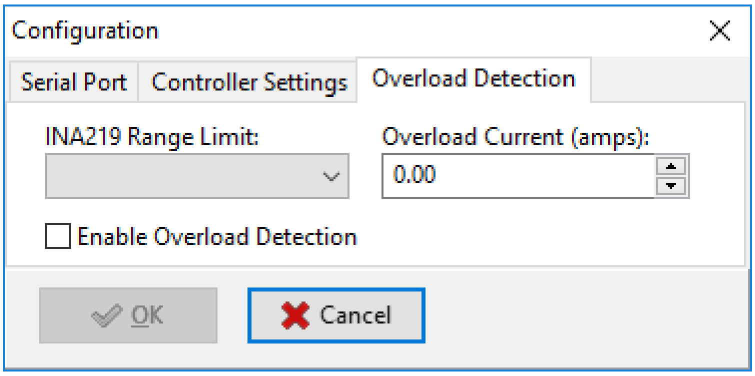
Enable Engine Locks

This option enables engine locking. This means no two serial interface controls can manage the same engine. If there are multiple users controlling trains then this should be enabled. If you are using a second serial interface for Bluetooth walk about control then you probably want this not enabled.

Enable Accessory Locks

This option enables accessory locking. This means no two serial interface controls can manage the same accessory address. If there are multiple users controlling points or accessories then this should be enabled. If you are using a second serial interface for Bluetooth walk about control then you probably want this not enabled.

The third page allows for the overload current detection to be configured:



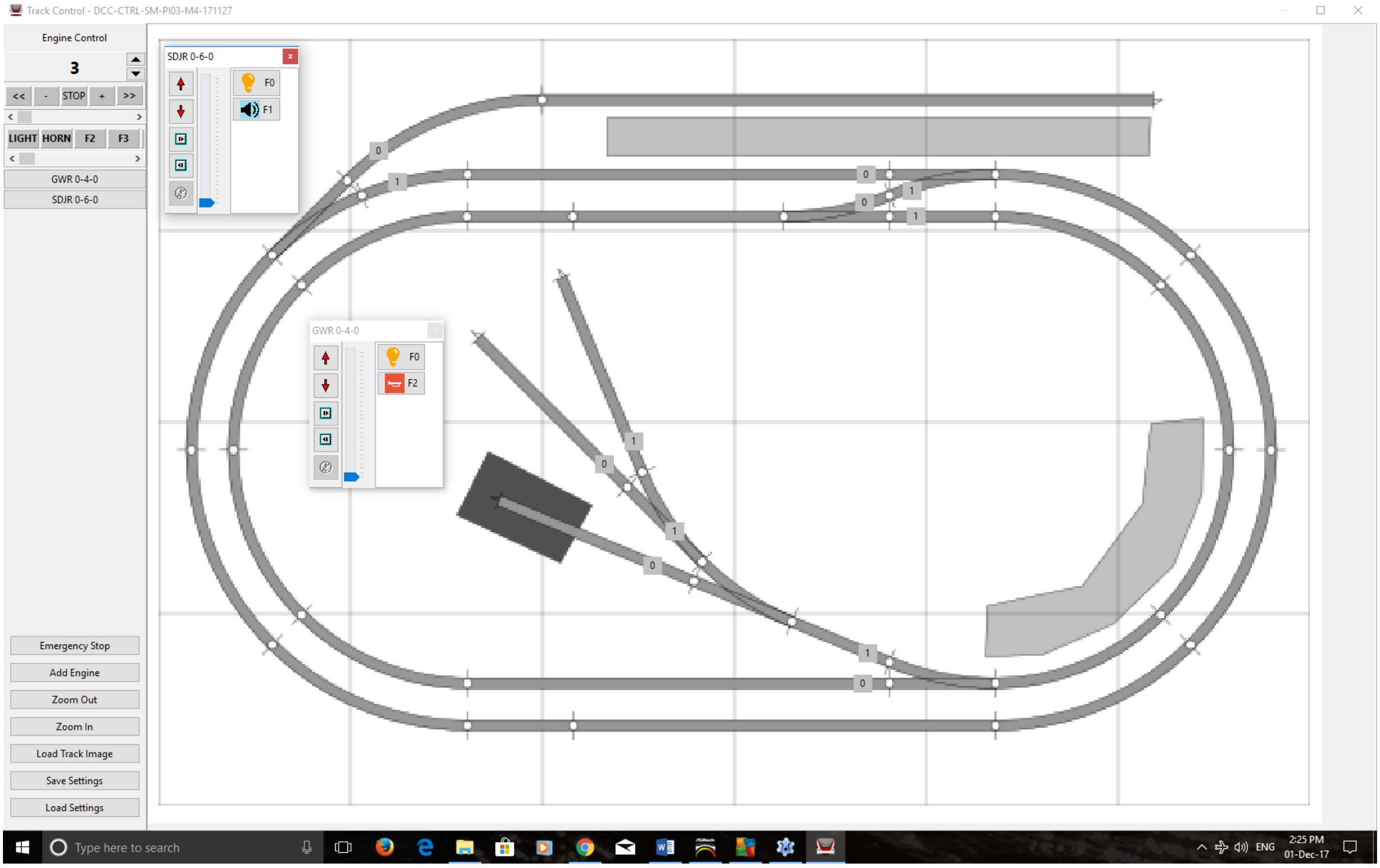
INA219 Range Limit:

Overload Current (amps): 0.00

□ Enable Overload Detection

The user must choose the INA219 range limit which can be: 3.2, 6.4, 9.6 or 12.8 Amps Max. This refers to the maximum current the INA219 can measure with the shunt resistor(s) attached. The current value at which the overload detection will disable the H Bridge is configured in amps using the second edit box under the **Overload Current (amps):** display. The overload detection is enabled by checking the **Enable Overload Detection** check box.

Track Control Window

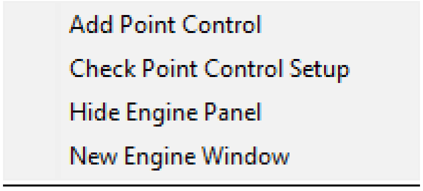


The track control screen allows control of both trains and points decoders. Each point can have two buttons allocated to it to control point direction. Engine buttons can be added to simplify engine control. Decoder function buttons can be labelled by the user for each engine address and the button on colour for each button can be defined. All configuration values can be saved and reloaded using the save/load settings buttons.

To use this screen an image of the track layout must be loaded using the **Load Track Image** button. This can be in any of the common image formats like .png, .jpg etc.

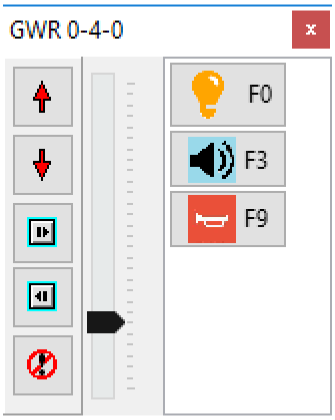
Popup Menu

The popup menu on the layout display allows addition of point control and engine windows. The user can also hide the left hand engine panel to maximize the track display and also check their point configuration.

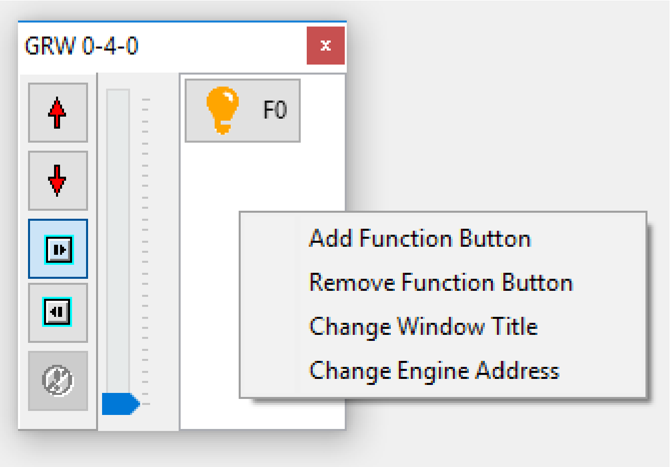


Adding Engine Windows

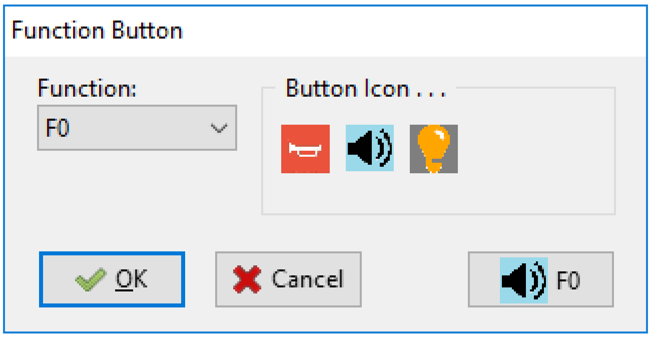
Engine windows can be added using popup menus from either the main track display or via engine buttons from the left hand panel. An engine form appears as:



New function buttons can be added by using the popup menu accessed via the right mouse click.



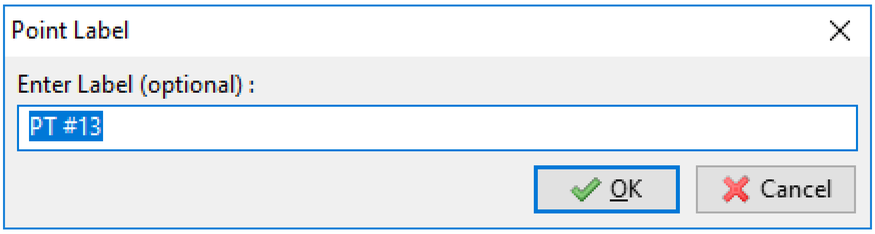
The engine address and window title can also be changed via the popup window. New function buttons are added via the following window:



The function F0-F28 is selected from the drop down list and an icon for the new function button is chosen from those displayed. An example of the new button is displayed in the bottom right corner of the window.

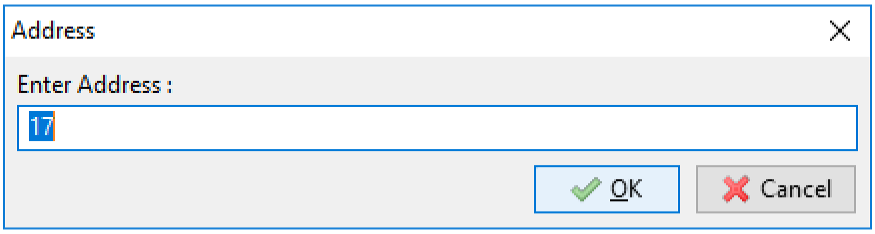
Adding Point Direction Buttons

To add point direction buttons use the right mouse button on the track image and then the popup menu **Add Point Control**, enter a label for the point and an address:



Point Label

Enter Label (optional) : PT #13



Address

Enter Address : 17

The point control setup can be checked by right clicking on the track image and selecting the **Check Point Control Setup** menu option. This will check addresses are not duplicated and direction values (0 or 1) are not duplicated. Any problems are highlighted red and yellow.

Once a point button has been added it can managed by using the popup menu for the point direction button. This is accessed using a right mouse click on the point direction button.

The popup menu has the following options:

Change Value (Direction)

This option allows the accessory value sent to the DCC unit to be changed, valid values are 0 to 7 inclusive. This value along with the address is used to switch a DCC point decoder when the button is clicked.

Change Address

This allows the DCC address used for the point direction button to be changed. When the point direction button is clicked this address along with the direction value 0 to 7 will be sent to the DCC control unit to control the point decoder.

Move

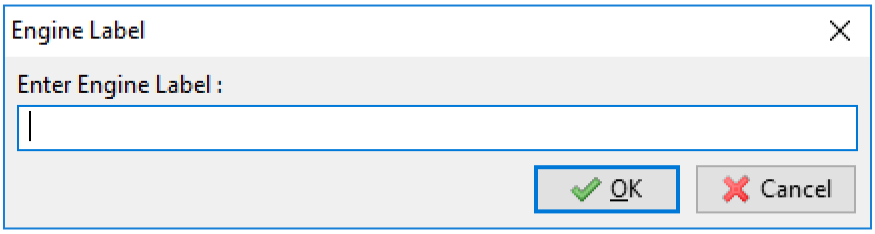
This option will move the point direction button with the mouse until the user clicks on the button or track image with the left mouse button.

Delete

This option allows the point direction button to be deleted.

Adding Engine Buttons

Engine buttons can be added using the **Add Engine** button. The button must have a label:



Engine Label

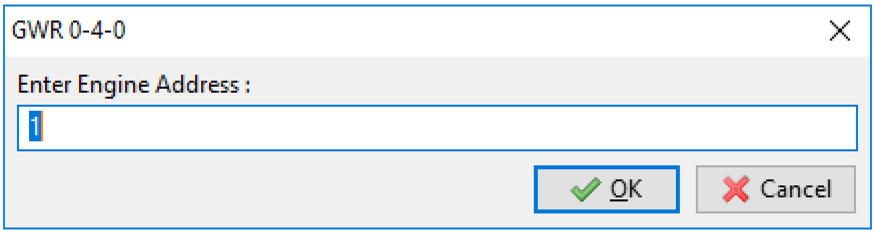
Enter Engine Label :

Each engine button has a popup menu accessed by using the right mouse click on the engine button.

The popup menu has the following options:

Change Address

The engine address can be changed using the following dialogue window:



GWR 0-4-0

Enter Engine Address :

1

Delete

The engine button can be deleted with this option.

Stop Engine

This option will send a DCC stop command for the engine address.

Engine Control

Apart from the engine buttons there is an engine address field that can be used to select engine address.

Below the engine address field are buttons to control engine speed + or -, engine direction >> or << for forward or reverse and an engine stop button.

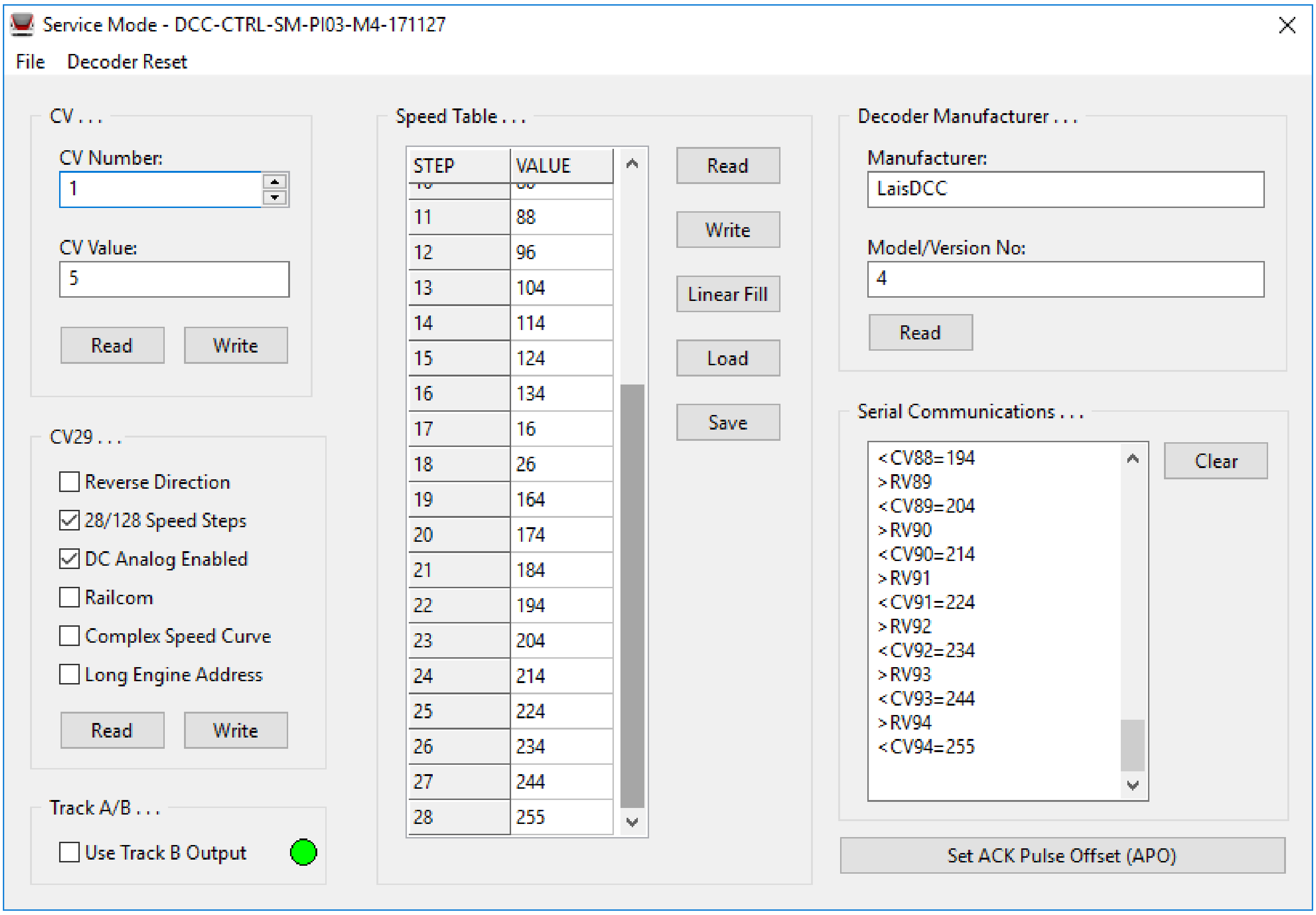
Below these buttons is a scroll bar that can also be used to set engine speed.

Below the speed scroll bar are buttons that allow engine functions to be turned on or off. The buttons currently support the NMRA DCC function group 1 functions F1..F4 and FL (light).

Function Buttons

Each function button has a popup menu accessed by using the right hand mouse button. The popup menu allows the function “on” colour to be changed and the button label to be changed for each engine address. All changes can be saved in the settings file.

Service Mode Window



CV Number: 1

CV Value: 5

□ Reverse Direction
☑ 28/128 Speed Steps
☑ DC Analog Enabled
□ Railcom
□ Complex Speed Curve
□ Long Engine Address

□ Use Track B Output ●

STEP	VALUE
11	88
12	96
13	104
14	114
15	124
16	134
17	16
18	26
19	164
20	174
21	184
22	194
23	204
24	214
25	224
26	234
27	244
28	255

Manufacturer: LaisDCC

Model/Version No: 4

```
<CV88=194
>RV89
<CV89=204
>RV90
<CV90=214
>RV91
<CV91=224
>RV92
<CV92=234
>RV93
<CV93=244
>RV94
<CV94=255
```

This screen allows decoder service mode programming of all CVs from 1 to 1024. For special CVs such as CV29 or the speed table there are special controls to simplify programming. For all other CVs there is a simple control to read or write individual CV values.

The DCC NMRA decoder ACK pulse is detected using the IN219 current monitor. The peak current before any service read is made is recorded and the peak current during the read is recorded. If the peak current increases this is considered to be a decoder acknowledge.

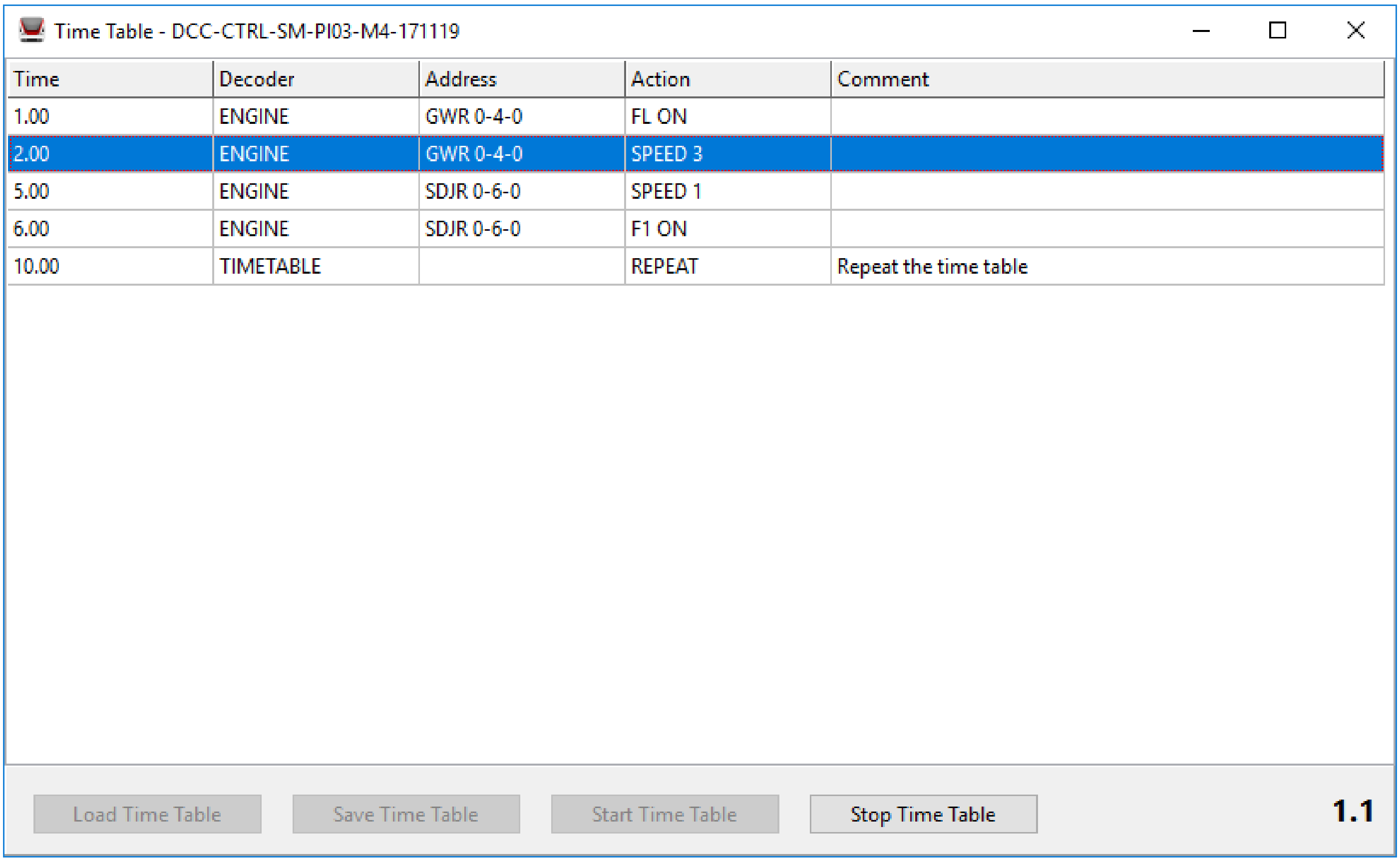
To overcome differences in decoders and noise on the current measurements the increase in current for an acknowledge may be modified using the APO setting. This adjusts the pulse threshold from 200 (5mA) to 800 (20mA) (the default is 440 about 11mA).

There is also a **Decoder Reset** menu which contains resets for various DCC decoders. A reset can however be achieved by using the CV write facility and following the decoder manufacturers reset instructions.

Track power can be turned on and off using the **File** menu options **Track Power Off** and **Track Power On**. A green or red LED in the **Track A/B** display shows track power status.

Time Table Window

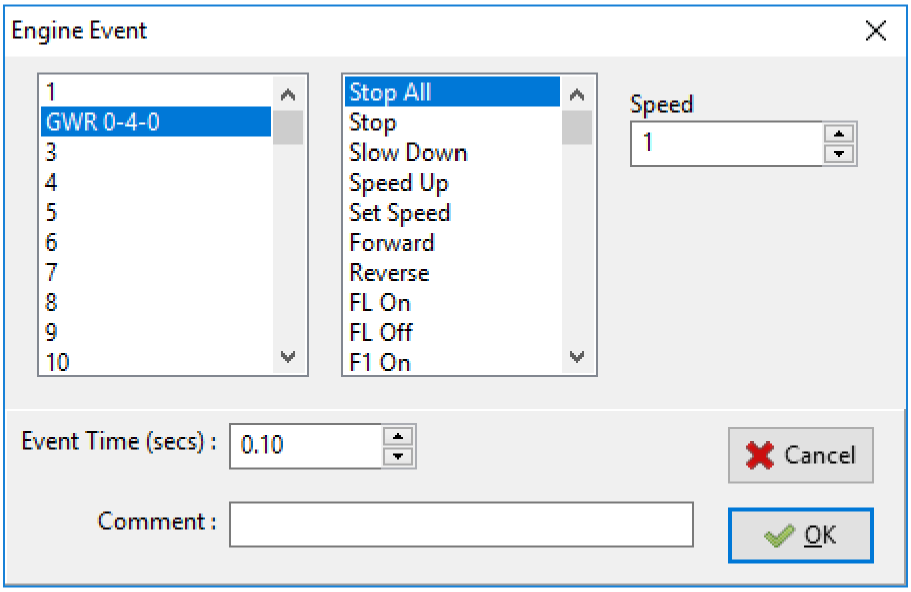
The time table window allows train function and accessory operations to be run from a time table. Engine speed and functions can be set as well as accessory on/off commands. The following is an example screen display of a running time table:



Time Table - DCC-CTRL-SM-PI03-M4-171119

Time	Decoder	Address	Action	Comment
1.00	ENGINE	GWR 0-4-0	FL ON	
2.00	ENGINE	GWR 0-4-0	SPEED 3	
5.00	ENGINE	SDJR 0-6-0	SPEED 1	
6.00	ENGINE	SDJR 0-6-0	F1 ON	
10.00	TIMETABLE		REPEAT	Repeat the time table

Using the right mouse button a menu can be accessed that allows adding, editing and deleting of time table events. The engine event form appears as below:



Engine Event

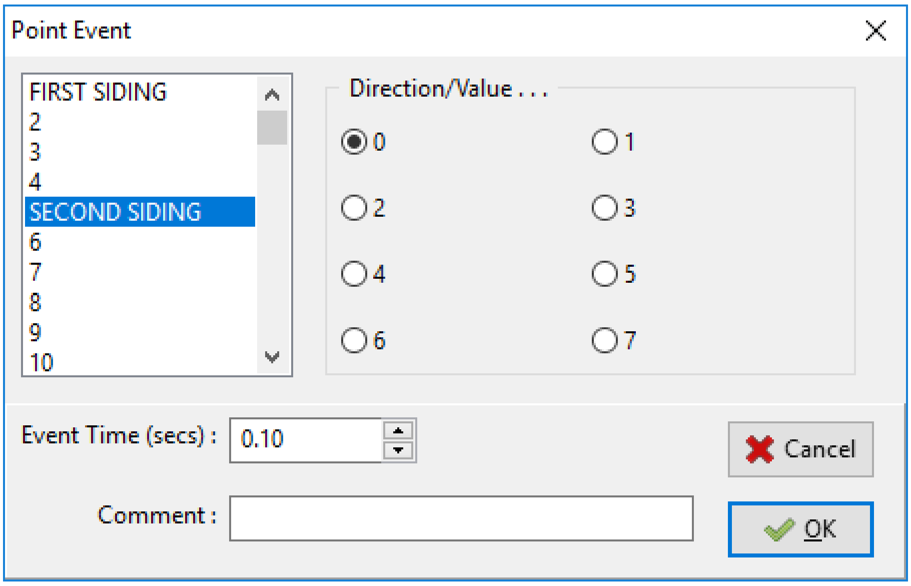
Speed: 1

Event Time (secs) : 0.10

Comment :

This form can be used to add or edit engine time table events. Engine addresses can be changed to engine names using the right mouse button to access a popup menu.

The accessory event form appears as below:



Point Event

FIRST SIDING
2
3
4
SECOND SIDING
6
7
8
9
10

Direction/Value . . .

● 0 ○ 1
○ 2 ○ 3
○ 4 ○ 5
○ 6 ○ 7

Event Time (secs) : 0.10

Comment :

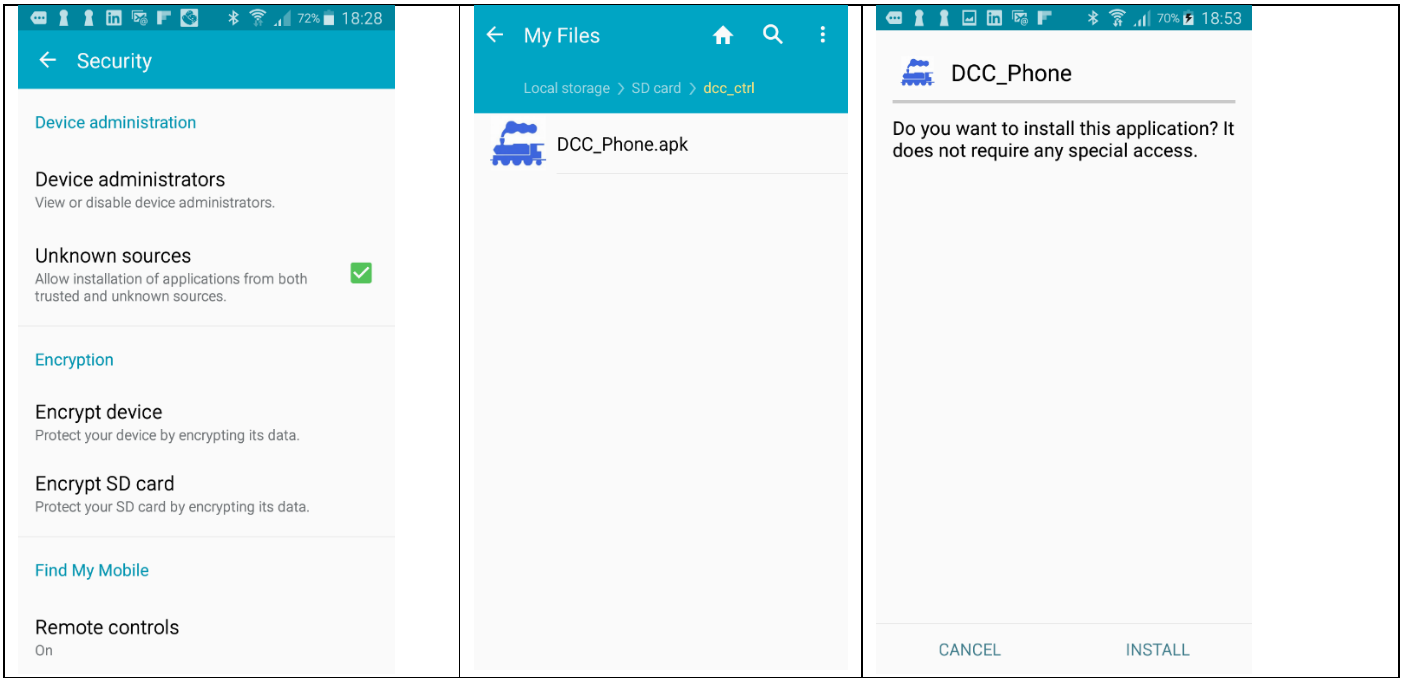
This form can be used to add or edit accessory (point) control events. Accessory addresses can be changed to meaningful names using the right mouse button to access a popup menu. Note any accessory value from 0 to 7 can be selected to be sent to the decoder. This supports all possible accessory modes available under NMRA DCC control.

Android Application

The Android application is stored in the .zip file as dcc_phone.apk. The application is installed onto the Android device using the following instructions:

Installing Android Application

To allow the software to be installed on the phone the security setting “unknown sources” must be enabled, see below. The file dcc_phone.apk is then copied to the Android device either by using USB or some other mechanism. The software is installed by running the “My Files” application, locating the dcc_phone.apk file copied to the phone and selecting it then choosing install.



← Security

Device administration

Device administrators
View or disable device administrators.

Unknown sources ☑
Allow installation of applications from both trusted and unknown sources.

Encryption

Encrypt device
Protect your device by encrypting its data.

Encrypt SD card
Protect your SD card by encrypting its data.

Find My Mobile

Remote controls
On

DCC_Phone.apk

DCC_Phone

Do you want to install this application? It does not require any special access.

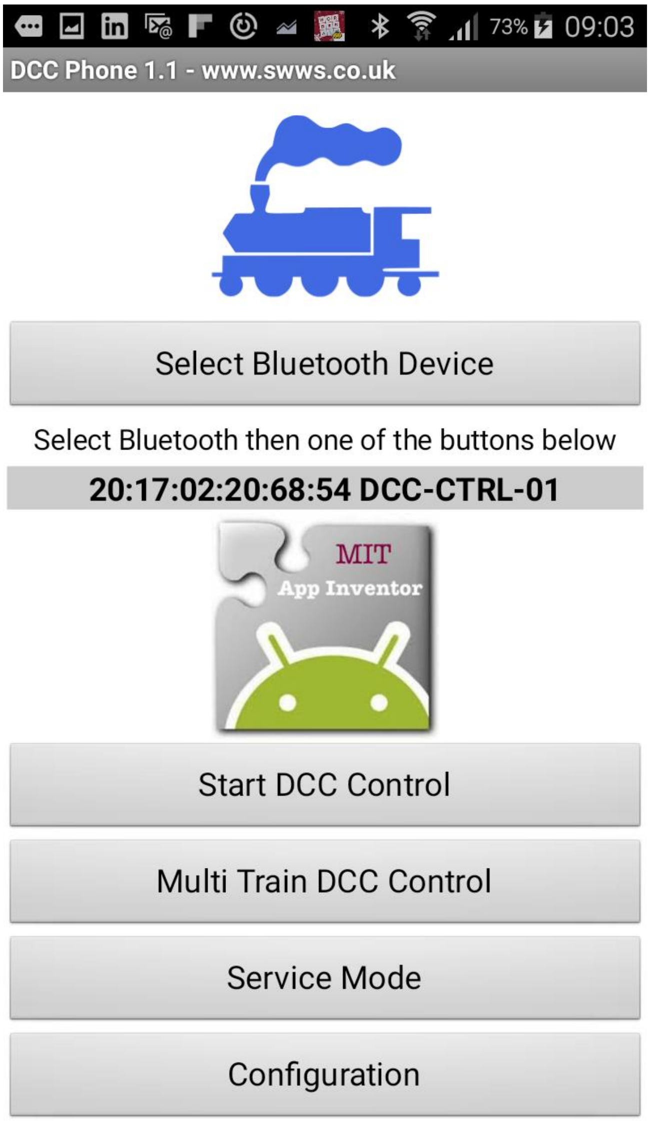
CANCEL INSTALL

Startup Screen

When the Android application is launched the screen shown below is displayed. The user must select a Bluetooth connection by clicking the **Select Bluetooth Device** button and choosing a Bluetooth connection. Once a connection has been chosen the other buttons become enabled and the user can select from the four main options:

- Start DCC Control
- Multi Train DCC Control
- Service Mode
- DCC Controller Configuration

These different screens are explained in the following sections.



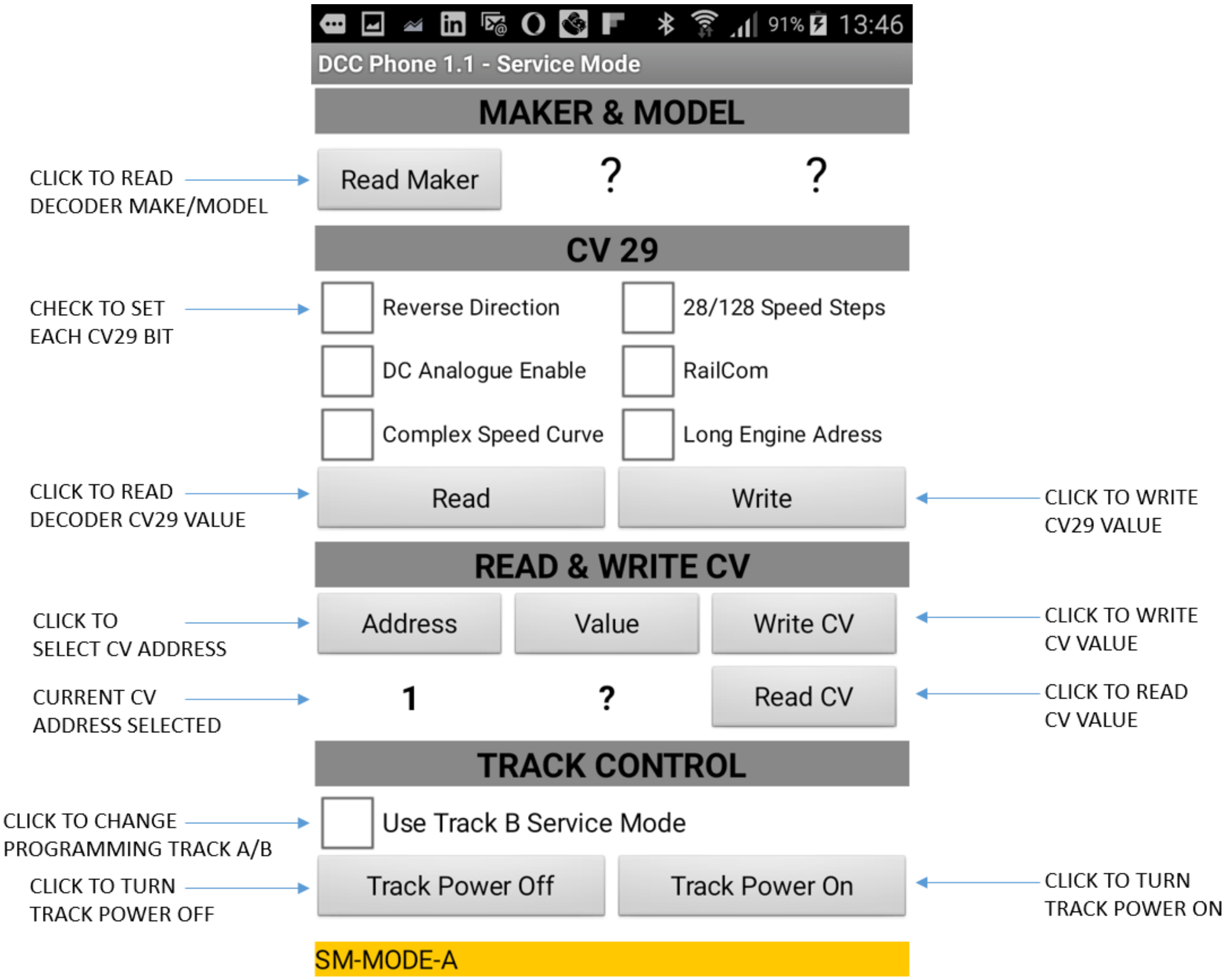
73% 09:03

DCC Phone 1.1 - www.swws.co.uk

Select Bluetooth then one of the buttons below

20:17:02:20:68:54 DCC-CTRL-01

Service Mode Screen



CLICK TO READ DECODER MAKE/MODEL

CHECK TO SET EACH CV29 BIT

CLICK TO READ DECODER CV29 VALUE

CLICK TO WRITE CV29 VALUE

CLICK TO SELECT CV ADDRESS

CLICK TO WRITE CV VALUE

CURRENT CV ADDRESS SELECTED

CLICK TO READ CV VALUE

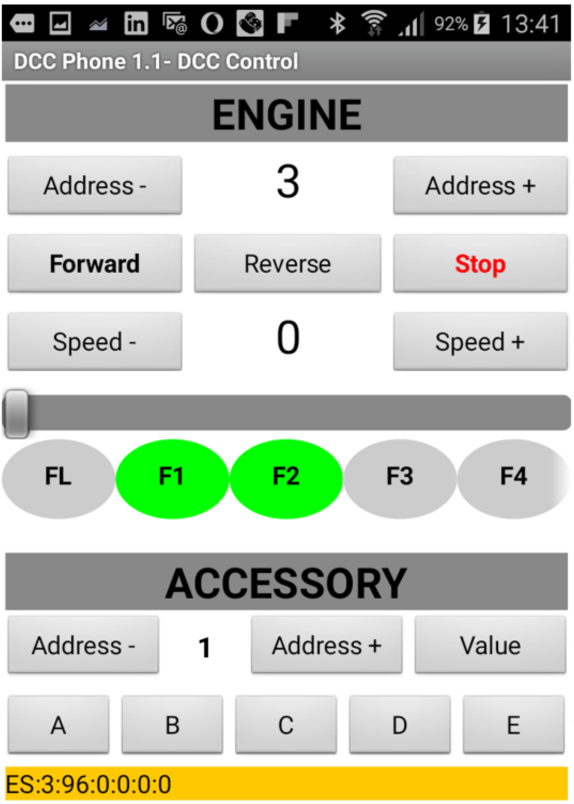
CLICK TO CHANGE PROGRAMMING TRACK A/B

CLICK TO TURN TRACK POWER OFF

CLICK TO TURN TRACK POWER ON

The service mode screen allows programming of decoder CV values. To read the decoder make and model click the **Read Maker** button. To program the CV29 value use the check boxes to set: reverse direction, 28/128 speed steps, analogue enable, railcom, complex speed curve and long engine addresses then click **Write**. To read CV29 click the **Read** button in the CV29 section of the screen. You can also use the **Read CV** and **Write CV** buttons for CV29 by entering the CV address as 29. To change between track A and track B use the **Use Track B Service Mode** check box. To read any CV address, select the address by clicking the **Address** button and then click **Read CV**. To Write any CV address, select the address in the address box, select the value by clicking the Value button and then click **Write**.

DCC Control Screen



ES:3:96:0:0:0:0

CLICK TO INCREASE
ENGINE ADDRESS

CLICK TO CHANGE
ENGINE DIRECTION

CLICK TO INCREASE
ACCESSORY ADDRESS
USE LONG CLICK FOR LIST

USE LONG CLICK TO
DISPLAY SELECTION LIST

CLICK TO STOP
ENGINE

CLICK TO INCREASE
ENGINE SPEED

CLICK TO TOGGLE
DECODER FUNCTION ON/OFF

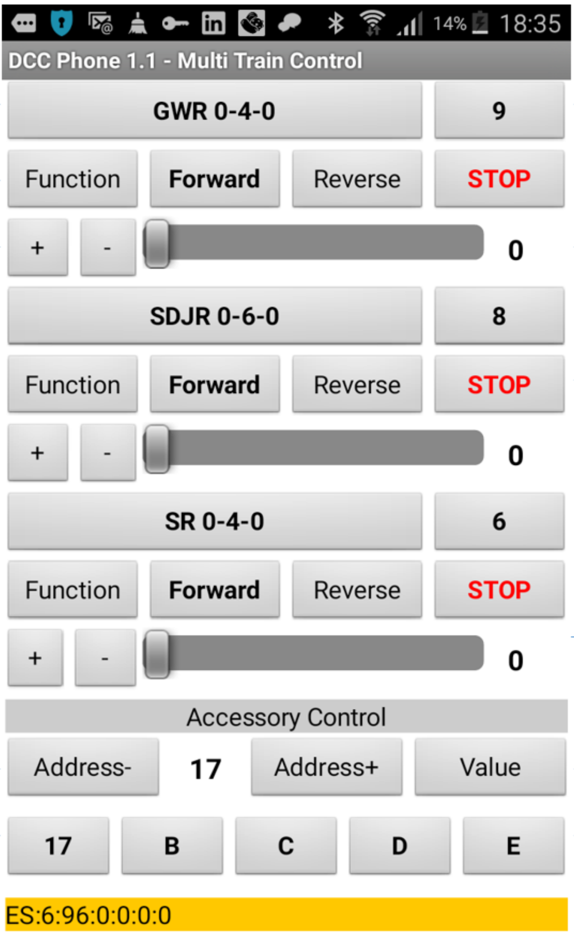
CLICK TO SEND
ACCESSORY VALUE

ACCESSORY SHORTCUT BUTTONS
CLICK/LONG CLICK FOR ADDRESS
THEN CLICK TO SET VALUE

The DCC control screen allows the control of engine decoders and accessory decoders. Use the **Address +** and **Address –** buttons to select an engine address. An **X** displayed in the engine speed display indicates that the engine is locked by another user. Once an unlocked engine address is chosen the engine speed will be displayed and the engine control buttons **Forward**, **Reverse**, **Stop**, **Speed –** and **Speed +** will become enabled, the engine speed slider will also become enabled. Use the FL (F0) to F28 buttons for the engine decoder functions. These buttons are highlighted light green when a decoder function is activated.

The accessory decoders can be turned on or off by using the **Address –** and **Address +** buttons to select an accessory address and then using the value button to send the accessory value 0 to 7. The address can also be selected by holding down either address button for a few seconds. There are also five accessory buttons **A**, **B**, **C**, **D** and **E**. These can used to set an accessory address. Once an address has been set pressing the button will turn on/off the accessory at the chosen address. To change an accessory address once set hold the button down for a couple of seconds.

Multi Train DCC Control Screen



ES:6:96:0:0:0:0

LONG CLICK TO CHANGE ENGINE LABEL

CLICK TO SEND ENGINE FUNCTION

CLICK PLUS OR MINUS TO CHANGE SPEED

CLICK PLUS OR MINUS TO CHANGE ACCESSORY ADDRESS

GREEN FONT INDICATES ACCESSORY ON (VALUE=1)

CLICK TO CHANGE ENGINE ADDRESS

CURRENT ENGINE SPEED X INDICATES ADDRESS LOCKED

CLICK TO STOP ENGINE

USE SLIDER TO CHANGE ENGINE SPEED

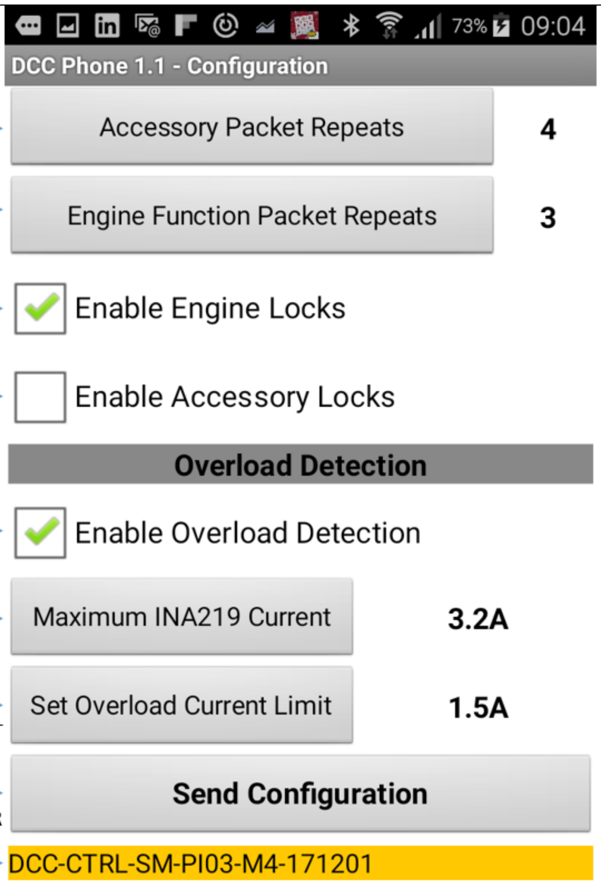
CLICK TO SEND ACCESSORY VALUE

CLICK A/B/C/D/E TO SET ACCESSORY ADDRESS THEN CICK TO CHANGE ACCESSORY VALUE

The multi train control screen allows the user to control three trains without having to keep changing addresses. Each train address is set by clicking the address button. Each train can also be given a label (name) by long clicking the engine label button. If an engine is already in use by another user then the engine speed display will show an **X** and no buttons will be enabled.

The accessory decoders can be programmed by using the **Address –** and **Address +** buttons to select an accessory address and then using the Value button to select a value to send to the decoder. There are also five accessory buttons **A**, **B**, **C**, **D** and **E**. These can used to set an accessory address. Once an address has been set pressing the button will turn on/off (send accessory value 0/1) the accessory at the chosen address. To change an accessory address once set hold the button down for a couple of seconds.

Configuration Screen



73% 09:04

DCC Phone 1.1 - Configuration

Accessory Packet Repeats 4

Engine Function Packet Repeats 3

☑ Enable Engine Locks

☐ Enable Accessory Locks

Overload Detection

☑ Enable Overload Detection

Maximum INA219 Current 3.2A

Set Overload Current Limit 1.5A

Send Configuration

DCC-CTRL-SM-PI03-M4-171201

CLICK TO CHANGE ACCESSORY PACKET REPEATS

CLICK TO CHANGE ENGINE PACKET REPEATS

CLICK TO ENABLE OR DISABLE ENGINE LOCKS

CLICK TO ENABLE OR DISABLE ACCESSORY LOCKS

CLICK TO ENABLE OR DISABLE OVERLOAD PROTECTION

CLICK TO SELECT MAXIMUM INA219 CURRENT

CLICK TO SET CURRENT PROTECTION LIMIT

CLICK TO SEND THE CONFIGURATION TO CONTROLLER

MESSAGE SENT FROM CONTROLLER

The configuration screen is used to configure:

Accessory Packet Repeats – The number of times an accessory packet is sent to an accessory decoder

Engine Function Packet Repeats – The number of times a function packet is sent to an engine (255=continuous)

Enable Engine Locks – When there are multiple users, lock an engine to a user

Enable Accessory Locks – When there are multiple users, lock and accessory to a user

Decoder Compatibility Table

Manufacturer	Model	Comments
LaisDCC	Engine Decoder 2 function 2A/1A bare wire	No known issues[1]
LaisDCC	Engine Decoder 4 function with NMRA 8 pin socket (860021)	No known issues[1]
Hornby	R8249 Engine Decoder	No known issues[1]
Hornby	R8247 Accessory Decoder	No known issues[1]
Gaugemaster	6 pin engine decoder	No known issues[2]
Hornby	TTS Decoder	No known issues[2]

1. These decoders have been tested by the DCC-CTRL developers.
2. These decoders are reported as working by users of the DCC-CTRL system.

Common Problems

CV Values are always read as zero in Service Mode

Check connections to programming track are good. Check power supply is on.

CV Values are read but appear incorrect

Check connections to programming track are good.

Engine moves during CV reading & writing

This is common as the engine powers the motor to signal acknowledge back to the DCC controller.

Hornby R8247 Decoder Not Working

These decoders can use four separate addresses for the four decoder point coil outputs. So when adding point controls to the track control layout you must add for each output two controls. One control direction must be set to 0 and the other set to 1 but they must have the same address. Also if the R8247 is programmed with a Hornby Select DCC controller, the address set on the controller will not be the DCC decoder address. For example if you program 61 on the Select controller this will program DCC decoder address 17 followed by 18, 19 and 20 for the four outputs.

If the R8247 is used in one address mode then the output ports are switched by setting an accessory decoder value of 0/1 for output 1, 2/3 for output 2, 4/5 for output 3 and 6/7 for output 4.

When programming decoder addresses using the DCC controller software remember to take one from the actual address you required before programming CV1 as the address range is 1..512 but the CV address range starts at zero (for address 1).

Website References

Low Cost DCC Controller Software

Email: support@swws.co.uk

L298N

http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/

INA219

https://www.adafruit.com/product/904

STM Nucleo F411RE

https://developer.mbed.org/platforms/ST-Nucleo-F411RE/

http://www.st.com/en/evaluation-tools/nucleo-f411re.html

http://www.st.com/en/embedded-software/stsw-link004.html

STM32F103

http://wiki.stm32duino.com/index.php?title=Blue_Pill

IBT-2

http://www.instructables.com/id/Motor-Driver-BTS7960-43A/

JMRI

http://jmri.org/

Low Cost DCC Controller with Service Mode Programming

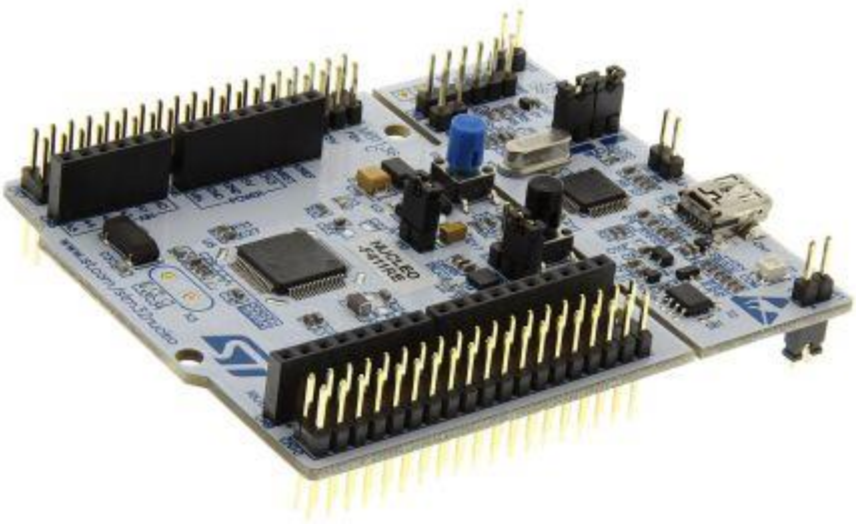
http://www.swws.co.uk/dcc/dcc_ctrl_serial_1.5.pdf

Sourcing Components

STM Nucleo F411RE

http://uk.rs-online.com/web/p/processor-microcontroller-development-kits/8224052/

Radio Spares £10.33 (account needed but free)



STM32F103

https://www.ebay.co.uk/itm/STM32F103C8T6-Cortex-M3-ARM-STM32-Minimum-System-Development-Board-Arduino-H53A/372113869631?epid=2162741928&hash=item56a3b70b3f:g:Q58AAOSwsBtZ7jaT

EBay £3.88 (lower price £1.65 from China)

L298N

http://www.ebay.co.uk/itm/Dual-L298N-H-Bridge-Stepper-Motor-Driver-Controller-Board-Arduino-Pi-ESP-UK-/182467585261?hash=item2a7bea64ed:g:iPMAAOSwax5YsZu4

Ebay £1.28 + 0.59 postage

IBT-2

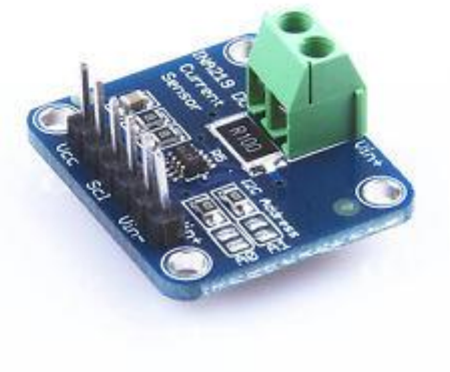
http://www.ebay.co.uk/itm/Semiconductor-Motor-Driver-Auto-BTS7960-43A-H-Bridge-PWM-Drive-For-Arduino-H7B8/112484915817?epid=1587309538&hash=item1a309f9a69:g:zKAAAOSw2tRZdvHX

EBay £10.74 (lower price £6.50 from Hong Kong)

INA219

http://www.ebay.co.uk/itm/LC-Technology-INA219-High-Side-DC-Current-Sensor-Module-R100-I2C-Flux-Workshop-/122359219862?hash=item1c7d2d8696:g:pB0AAOSw4A5Ypsz0

EBay £5.79 (lower price £1.67 from China but you need to solder some connectors)



Connecting Wires

http://www.ebay.co.uk/itm/40pcs-10cm-Dupont-Jumper-Wire-Ribbon-GPIO-Cable-PiArduino-Breadboard-F-F-M-M-F-M-/231983269314?var=&hash=item360347c5c2:m:mZQADiCEcb3mvuylcnJ5-RA

EBay £1.20 (female to female)

USB to TTL FT232RL FTDI

EBay £2.60 (lower price £1.67 from China)

Version Change History

1.7

Added new decoder reset support on Windows service mode screen. Added accessory value support for all values (0 to 7) on Android application “quick” accessory buttons.

1.6

Minor bug fixes only.

1.5

Added multi-train control. Added multi-user support for up to three users at once. Added 10amp H bridge option. Updated Android application to 1.1 version. Added current overload detection. Added track power control. Added STM32F103 processor option.

1.4

Added NMRA DCC functions F0 (FL) to F28 support. Fixed service mode problem when powering +5V STM32F411RE from L298N. Added Android application 1.0 version.

1.3

Added Blue Tooth support. Added Time Table window.

1.2

Minor bug fixes.

1.1

Added new track control window.

1.0

Initial version.