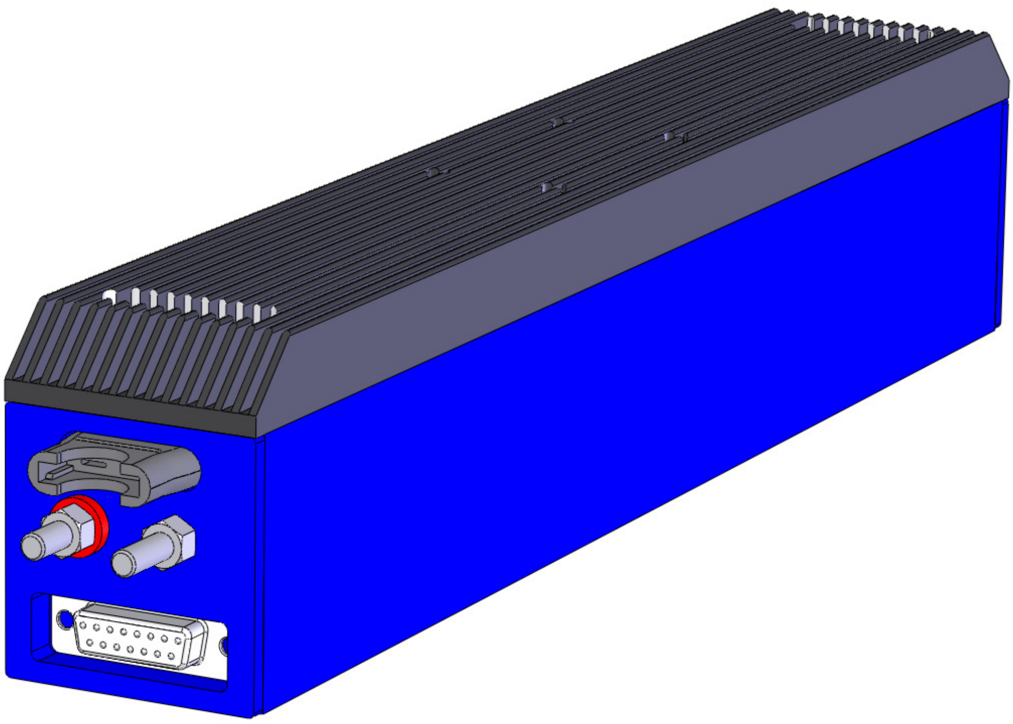
User Manual

Scimar Engineering Ltd

Split-Pi UDIO.60.25.L



Split-Pi is manufactured under licence from 3D Instruments Ltd by Scimar Engineering Ltd and protected by patent No. GB2376357 and worldwide equivalents.

MAN-UDIO6025L

Split-Pi UDIO.60.25.L
Edition: September 2010
Document Reference: MAN-UDIO6025L

IMPORTANT

READ ME FIRST

There are some important details to know and do's and don'ts to follow before using Split-Pi:

- Split-Pi has some features that do not exist in other systems, and controls a lot more power than other devices of this size. Therefore damage can easily occur to connected equipment without a good understanding of how Split-Pi operates.

Whilst Split-Pi is 100% symmetrical, the internal control is NOT. It implements a simple voltage ratio control which works by reference to the voltage on the Left Hand side terminals.

- DO Read the ‘Hardware Orientation’ section in the User Guide to establish the 'Handedness' of Split-Pi.

Devices that the user connects to Split-Pi must be able to sustain current flow in the directions that the user commands by varying the voltage ratio. The GENERAL case is given below

Left-Hand Side (LHS)	Right-Hand Side (RHS)
Primary Power Source eg rechargeable battery or other power source capable of reverse current flow such as a battery or capacitor bank	**Primary Load** eg DC motor or Secondary Power Source

- DO NOT pass current into power sources that are not rechargeable.
- If a secondary power source is used on the RHS then the user should make provision for external switching to enable that source to be connected or disconnected.
- DO balance the voltage on the Right-Hand Side of Split-Pi with the voltage of the secondary power source before connecting it

The internal control inside Split-Pi requires power to function. Internal diode connections route this from both the LHS and/or RHS main terminals so this supply can come from either dynamically, but this supply must be 15V or more.

- In operation Split-Pi must have the voltage at one side or the other above 15V at all times.

- The maximum working ratings for the device are 60V +/- 25 Amps at any terminal. It is possible to command an output voltage greater than 60V, and it is the user's responsibility to remain inside these limits.

- 30 Amp fuses are fitted because internal dynamic current limiting protects the device in normal operation.

- DO NOT fit fuses rated above 30Amps.

Contents

Specification

Topology

Split-Pi is the name given to a novel, patented DC-DC converter technology, which is shown in its most generic form in Figure 1.

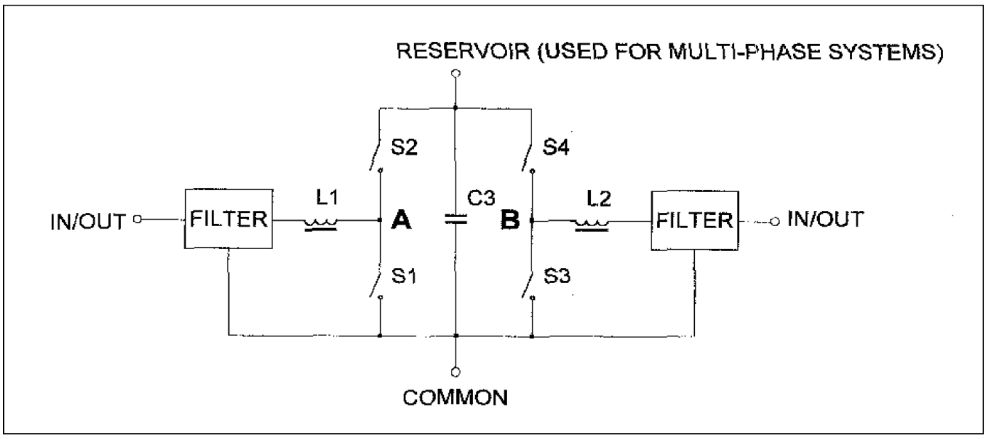


Figure 1- Split-Pi Patented Topology

Switches S are implemented in Power MOSFETs. The device is entirely symmetrical, and can both source and sink current, and convert both up and down from a given supply voltage, and for this reason the acronym UDIO (Up-Down, In-Out) is applied. In a MOSFET implementation the device is fully 'synchronous'. The absence of any diodes conducting means that there are no irreducible voltage drops and so efficiency can approach 100% very closely.

Control

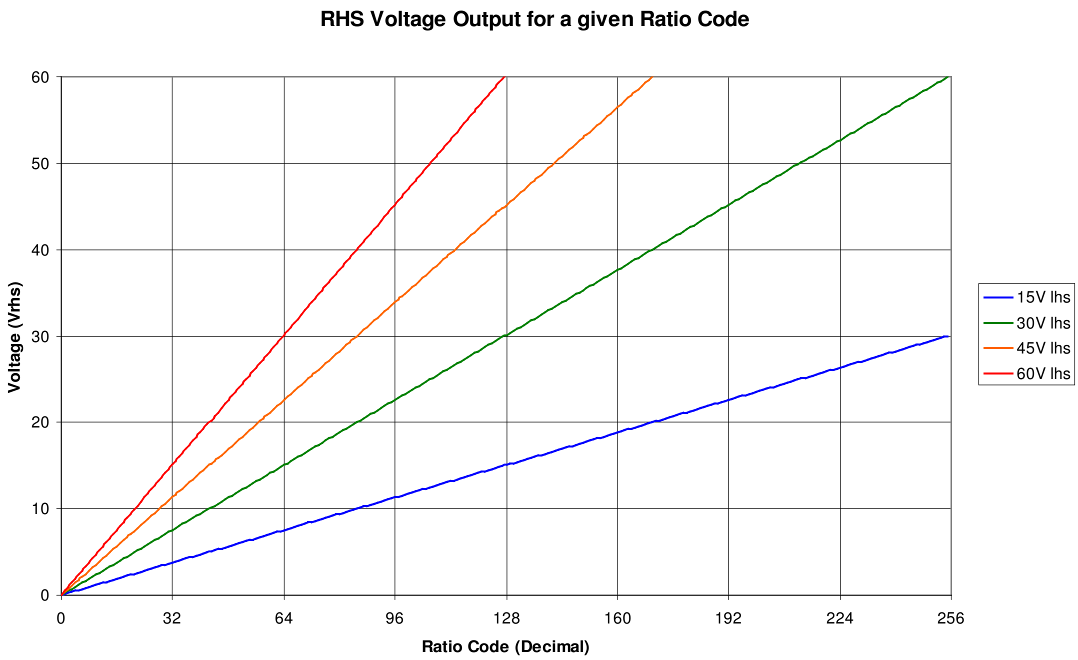
The control circuitry on-board Split-Pi UDIO.60.25.L uses one side of the device as a voltage reference to control Split-Pi which is referred to as the “Left-Hand Side”. This provides the user with a simple to follow convention:

- Left-hand side of Split-Pi
    - Input of device
    - Connected to a power source (battery)
- Right-hand side of Split-Pi
    - Output of device
    - Connected to the equipment requiring control (motor)
- Positive current from Left to Right

The right-hand side voltage is governed by a 256-code (0-255), real-time, programmable ratio. The right-hand side voltage can be calculated using the equation:

$$V_{rhs} = V_{ratio} \frac{V_{lhs}}{128}$$

The following chart illustrates the operational right-hand side output range of Split-Pi with different left-hand side input voltages:



Since Split-Pi is symmetrical, allowing current in and out, voltage up and down, situations can occur where power transfer is in the reverse direction (right to left) such as regenerative braking by a right-hand side motor recharging the left-hand side supply battery. Due to its left-hand side controller reference, Split-Pi should never be connected in a reverse fashion with battery/supply on the right as this can damage

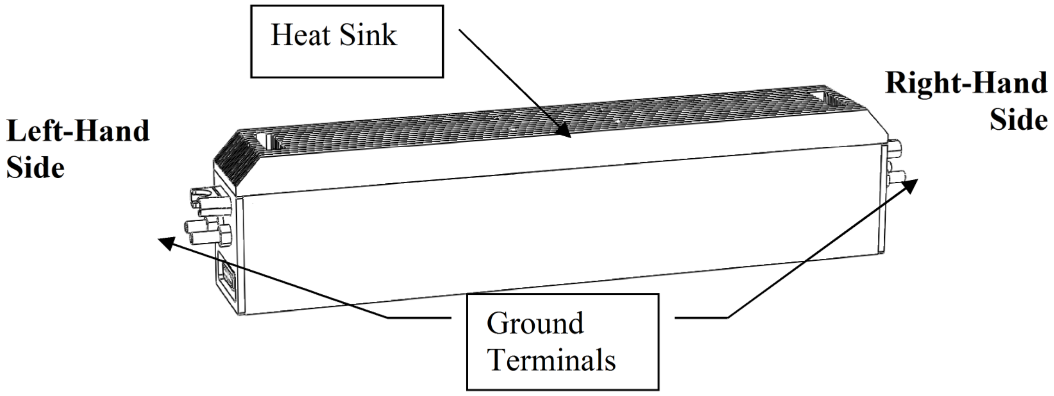
Split-Pi and any connected equipment. Table 1 shows the input and output ranges of Split-Pi in positive and negative current directions.

Table 1 - Split-Pi IO Range

Parameter	Value	Unit	Conditions
Positive Current (Left to Right)			
Left-Hand Side Input			
LHS Voltage min	15	V	
LHS Voltage max	60	V	
LHS Current max	25	A	
Right-Hand Side Output			
RHS Voltage min	0	V	
RHS Voltage max (2 x V_{lhs})	60	V	LHS Voltage ≥ 30V
RHS Current max	25	A	
Reverse Current (Right to Left)			
Left-Hand Side Output			
LHS Voltage min (0.5 x V_{rhs})	7.5	V	RHS Voltage = 15V
LHS Voltage max	60	V	
LHS Current max	25	A	
Right-Hand Side Input			
RHS Voltage min	15	V	
RHS Voltage max	60	V	
RHS Current max	25	A	

Hardware Orientation

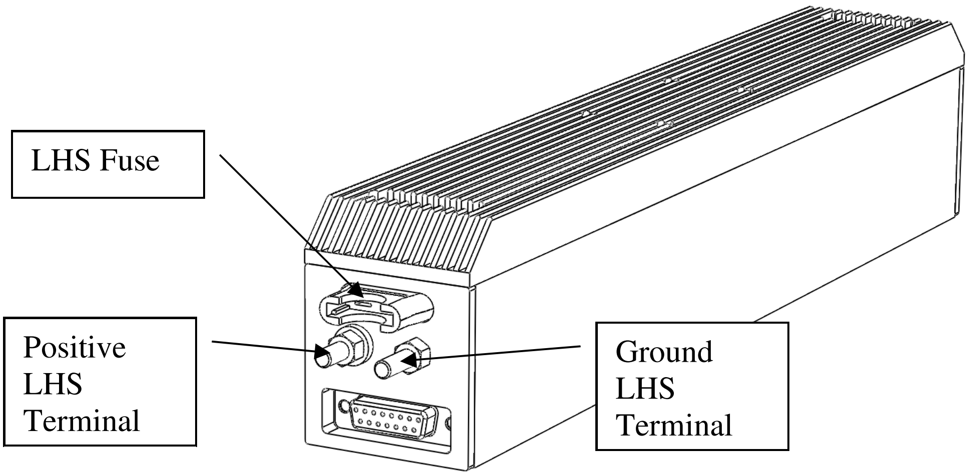
It is important to familiarise yourself with the handedness of Split-Pi. Although Split-Pi is symmetrical, its' control circuitry is referenced to the left-hand side for ease of use. Incorrectly connecting power to the right-hand side can cause damage to Split-Pi and any connected equipment.



With the heat sink at the top, position Split-Pi so that the ground terminals on both sides are closest to you. Ground terminals can be identified by having black or no plastic washers fitted. In this orientation the left-hand side of the device is naturally to your left and the right-hand side is to your right.

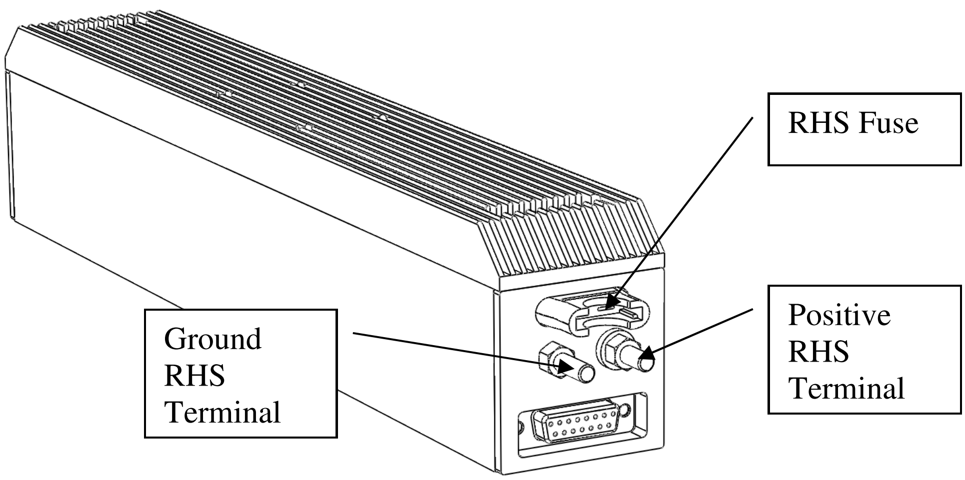
Left-Hand Side

The left-hand side of Split-Pi is the input side and should be connected to the DC supply equipment such as batteries.



Right-Hand Side

The right-hand side of Split-Pi is the output side and should be connected to the equipment you wish to drive such as a DC motor.



Data Connection

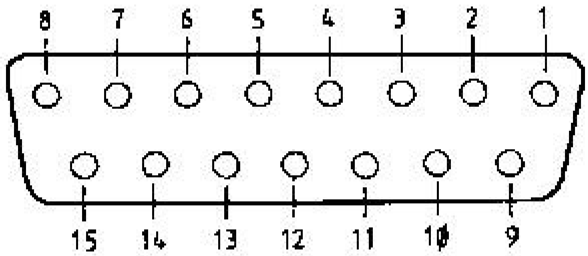


Illustration 1: Female 15way D-Type

Pin	Signal
1	NC
2	NC
3	NC
4	NC
5	NC
6	NC
7	NC
8	NC
9	Iloop -
10	Iloop +
11	NC
12	NC
13	NC
14	NC
15	NC

Table 2: Connector Pinouts

iLoop Overview

The iLoop communication featured in Split-Pi devices is an isolated communication device that uses flow of current to send and receive data. The iLoop communication protocol can be split into standardised layers as defined by the OSI model.

Physical Layer

The iLoop transceiver in Split-Pi devices is a slave device that senses the flow of current to receive data and switches the current on and off to send data. A single iLoop master device must be used to supply power to the current loop. Currently there is an RS232 – iLoop adapter available that supplies current to the current loop and provides a standard RS232 interface for simple connection to a computer. Below is a simplified illustration of the master iLoop circuit with it's constant current source and isolated transmitter and receiver. The master iLoop schematic is available for

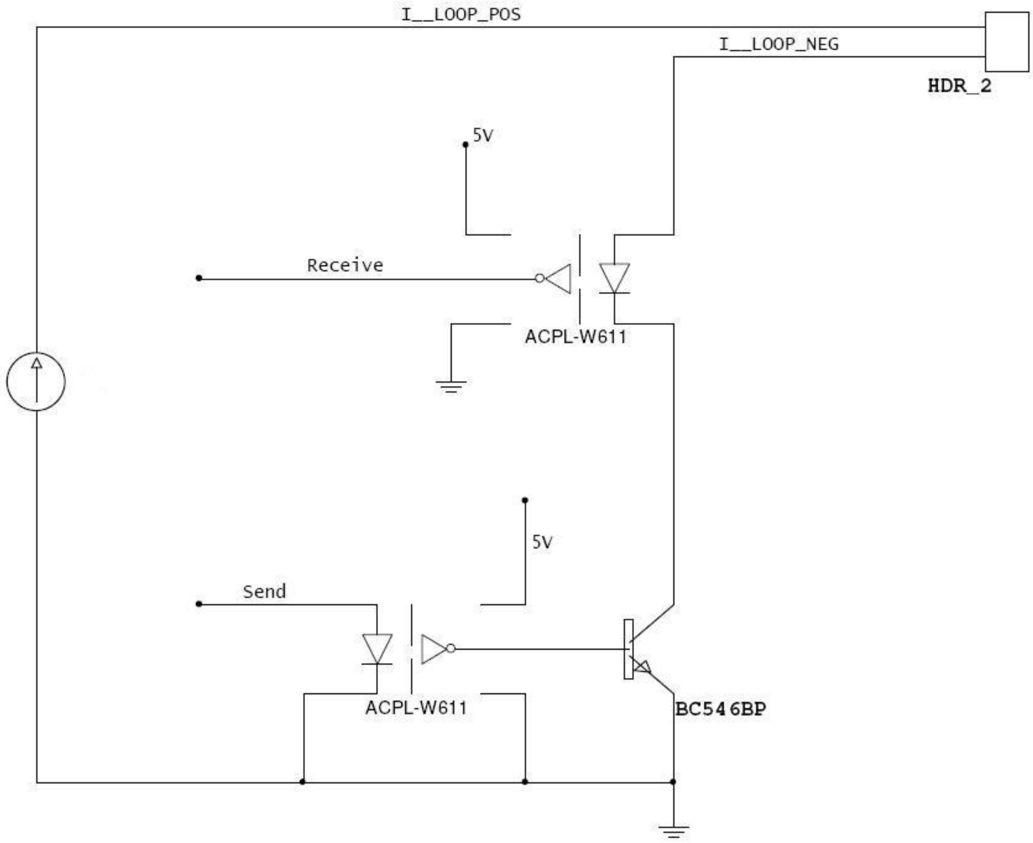


Illustration 2: Simplified Master iLoop Circuit

integration into an embedded control board or other system. The slave iLoop circuitry in Split-Pi devices is very similar to the master with the omission of the constant current source.

The physical communications uses only two wires; iLoop+ and iLoop- to send and receive data asynchronously in half duplex mode. The communication works at 19200 bits per second, sends 8 data bits in a frame, has no parity bit, has one stop bit and has no flow control. It functions in a similar way to RS232, apart from the signals are carried by current flow rather than voltage levels.

Data Link Layer

The iLoop protocol uses addressing in a similar way to IEEE 802.3 Ethernet MAC to communicate with several Split-Pi devices connected to the same iLoop. It has three addressing modes; Direct, Group and Broadcast. Where direct addressing communicates with a single Split-Pi device, group addressing with a group of Split-Pi device connected to the same iLoop and Broadcast with every Split-Pi device connected to that iLoop. This allows for multiple groups of Split-Pi devices connected to the same controller, for instance a group of Split-Pi's for each wheel on a vehicle, allowing individual wheel speed control.

Network Layer

To begin a communication transaction, an eight bit addressing mode word is transmitted on the iLoop to specify whether one or many devices will act on the communication. To perform direct addressing send the 8 bit word 0x4E (ASCII character 'N'). To perform group addressing send the 8 bit word 0x47 (ASCII character 'G') To broadcast to all devices, either Direct or Group addressing can be used as it uses a broadcast address.
Then an eight bit address will be transmitted to specify the Split-Pi device/s that should further act on the communication. As default all Split-Pi devices come programmed with the address 0xFE. The broadcast address is 0xFF.
Then an eight bit command word followed by variable length arguments specifying the action that the Split-Pi device/s should take.
Some commands are then followed by a variable length response from a Split-Pi device.

The message structure and length is shown below:

8 bits	8 bits	8 bits	8 – 32 bits	8 – 32 bits
Addressing Mode	Address	Command	Argument	Response

Table 3: iLoop Message Frame

Transport Layer

The commands are summarised in the table below:

Command	Description	Command ASCII (Hex)	Address Mode	Argument Bytes	Response Bytes
SetCode	Set the node(s) voltage ratio code.	V (0x56)	B	1	
Hotstart[1]	Set the node(s) voltage ratio code to hotstart	X (0x58)	B	1	
GetCode	Get the node voltage ratio code	O (0x4F)	D		1
GetStatus	Get the node status	S (0x53)	D		2
ReadADC	Read the nodes specified ADC channel	I (0x49)	D	1	2
ReadEEPROM	Read a byte from a specified address within the nodes EEPROM	Y (0x59)	D	2	1
Halt	Go open circuit, halt voltage conversion.	H (0x48)	B		
SetGID	Set the node(s) group ID and store in EEPROM	J (0x4A)	G	1	
GetGID	Get the nodes currently active group ID	P (0x50)	D		1
SetNID	Set the full node ID and store in EEPROM	T (0x54)	D	4	
GetNID	Get the currently active full node ID	W (0x57)	D		4
RefreshConf	Refresh the node configuration from EEPROM	U (0x55)	G		
ResetDefault	Revert to default (Factory) configuration	? (0x3F)	D		

Table 4: iLoop Command Set

Note 1: Hotstart requires special care and attention in its use. Please see iLoop Manual for more info.

For example to set a voltage ratio of 64 (40 hex) using direct addressing:

NFEV40

To set a voltage ratio of 64 using group addressing:

G55V40

To read the currently set voltage ratio (interrogation only uses direct addressing):

NFEO

Split-Pi will then return a two-character Ascii Hex value of the code, i.e. 40

Calibrated Analogue Readings

Split-Pi has a built in Analogue to Digital converter that allows the user to read Voltages applied to the left and right hand sides, bi-directional Currents at either sides and the temperature of the synchronous switching bridges. The values read from the ADC are absolute values and require conversion into real values. This is achieved by reading calibration data stored in the EEPROM and running it through the equation:

$$Actual\ Value\,(10^{-6}) = M\,[\,ADC - (2^{(ADC_{bits}-16)} \times C)]$$

where M and C are 16 bit values stored in EEPROM, ADC_{bits} is an 8 bit value stored in EEPROM and ADC is the Analogue to Digital reading. M is the actual value of one count of the ADC in microVolts, microAmps, micro-degC. C is the zero point offset in ADC counts.

Use the ReadEEPROM command 'Y' (0x59) to read the calibration data from Split-Pi's EEPROM along with the following table of addresses. The highlighted locations are the most revelant:

Location (Hex)	Name	Location (Hex)	Name	Location (Hex)	Name
0x00	Address 0	0x10	Volt Left M0	0x20	Current Left M0
0x01	Address 1	0x11	Volt Left M1	0x21	Current Left M1
0x02	Address 2	0x12	Volt Left C0	0x22	Current Left C0
0x03	Address 3	0x13	Volt Left C1	0x23	Current Left C1
0x04	Address Length	0x14	Volt Right M0	0x24	Current Right M0
0x05	Group Address	0x15	Volt Right M1	0x25	Current Right M1
0x06	Control State	0x16	Volt Right C0	0x26	Current Right C0
0x07	Upper Limit	0x17	Volt Right C1	0x27	Current Right C1
0x08	Lower Limit	0x18	NA	0x28	Temp Left M0
0x09	Firmware Version	0x19	NA	0x29	Temp Left M1
0x0A	Calibration Version	0x1A	NA	0x2A	Temp Left C0
0x0B	Serial Number 0	0x1B	NA	0x2B	Temp Left C1
0x0C	Serial Number 1	0x1C	NA	0x2C	Temp Right M0
0x0D	Serial Number 2	0x1D	NA	0x2D	Temp Right M1
0x0E	Serial Number 3	0x1E	NA	0x2E	Temp Right C0
0x0F	ADC Bits	0x1F	NA	0x2F	Temp Right C1

Table 5: Main EEPROM Mapping

For example to read the ADC Bits (only accessible using direct addressing):

NFEYA00F

Split-Pi will return two ASCII characters for example, '0C' in ASCII Hex converted to a decimal value this is 12.
Each of the values stored in EEPROM are 8bits. Two bytes must be combined to get the 16bit value of M or C where M = [M1:M0] and C = [C1:C0].

Once you have the calibration data you will need to read the ADC using the ReadADC command 'I' (0x49) . The following table shows the ADC channels:

Channel	ASCII Hex Value	Description
0	00	Voltage Left
1	01	Voltage Right
2	02	NA
3	03	NA
4	04	Current Left
5	05	Current Right
6	06	Temperature Left Bridge
7	07	Temperature Right Bridge

Table 6: ADC Channels

For example to read the left hand side current (direct addressing only):

NFEI04

Split-Pi will return four ASCII characters representing a 16 bit value, for example '096F' which converted to decimal is 2415.

To convert this to a human readable value run this through the previous equation with the related M and C values for left hand side current. For example:

Left Current M0 at location 0x20 = 0x62
Left Current M1 at location 0x21 = 0x51
therefore M = [M1:M0] = 0x5162 = 20834 (decimal)

Left Current C0 at location 0x22 = 0x30
Left Current C1 at location 0x23 = 0x80
therefore C = [C1:C0] = 0x8030 = 32816 (decimal)

ADC Bits at location 0x0F = 0x0C = 12 (decimal)

and as in the previous example Left Current Channel 0x04 reads 0x096F = 2415

now put them into the equation:

$$Actual\ Value\,(10^{-6}) = M\,[ADC - (2^{(ADC_{bits}-16)} \times C)]$$

$$Current\,(\mu A) = 20834\,[2415 - (2^{(12-16)} \times 32816)]$$

$$Current = 20834[2415-(0.0625 \times 32816)]$$
$$Current = 20834[2415-2051]$$
$$Current = 20834[364]$$
$$Current = 7583576 \mu A$$
$$Current \approx 7.584 A$$

Appendices

ASCII Tables

Hex	30	31	32	33	34	35	36	37	38	39
Dec	48	49	50	51	52	53	54	55	56	57
ASCII	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Table 7: ASCII Numbers

Hex	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
Dec	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
ASCII		**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
Hex	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
Dec	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
ASCII	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**					

Table 8: ASCII Letters