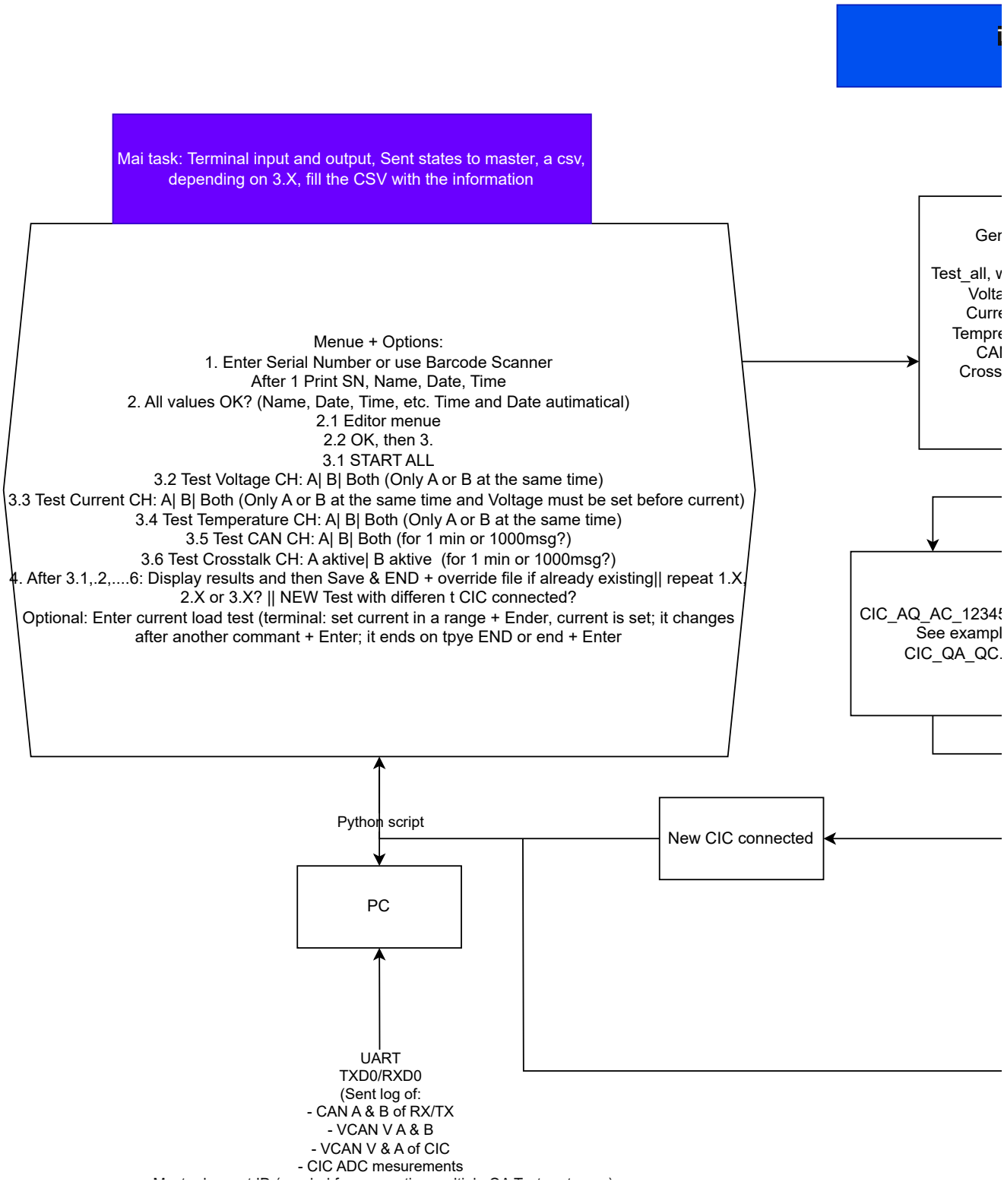
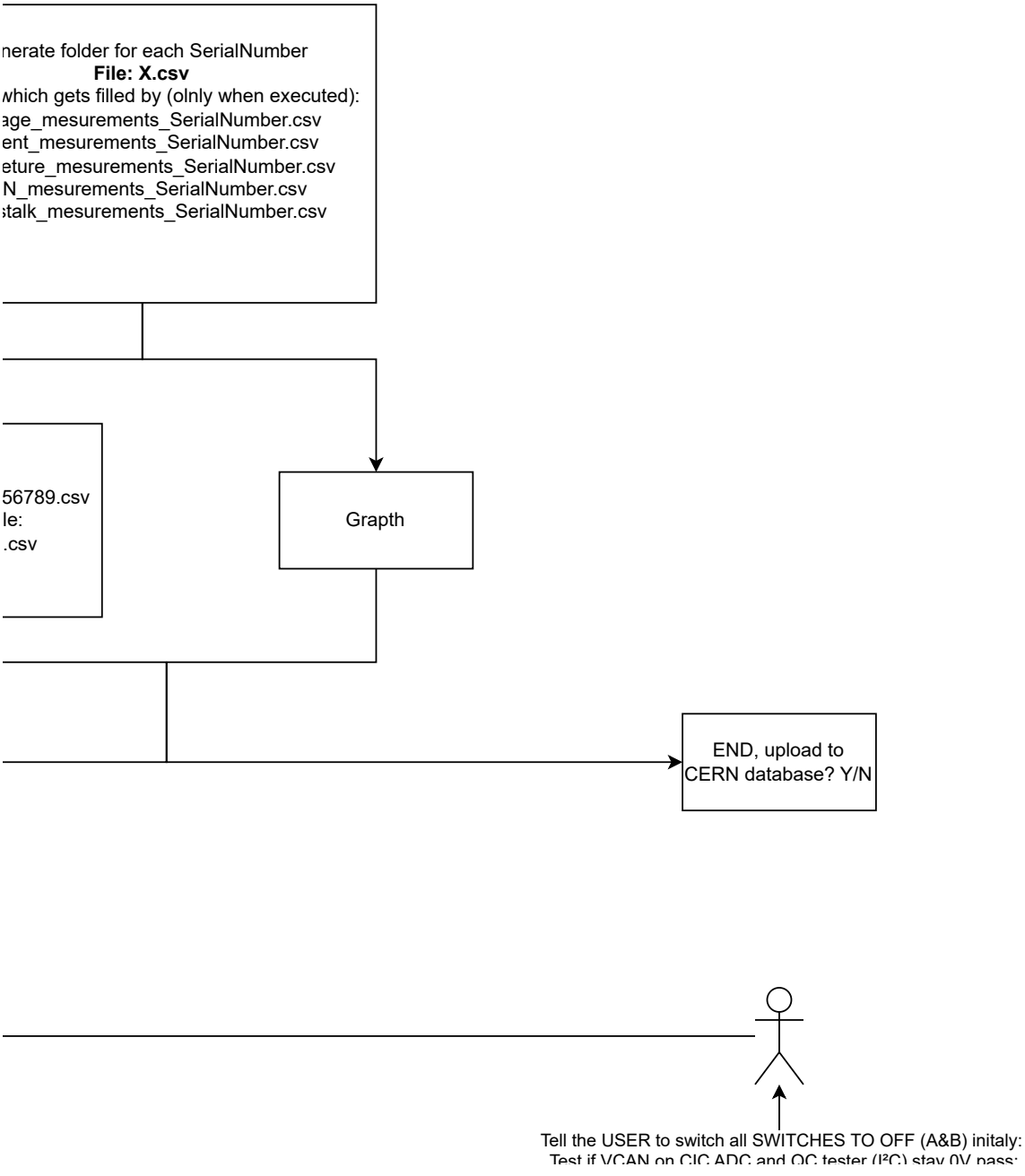
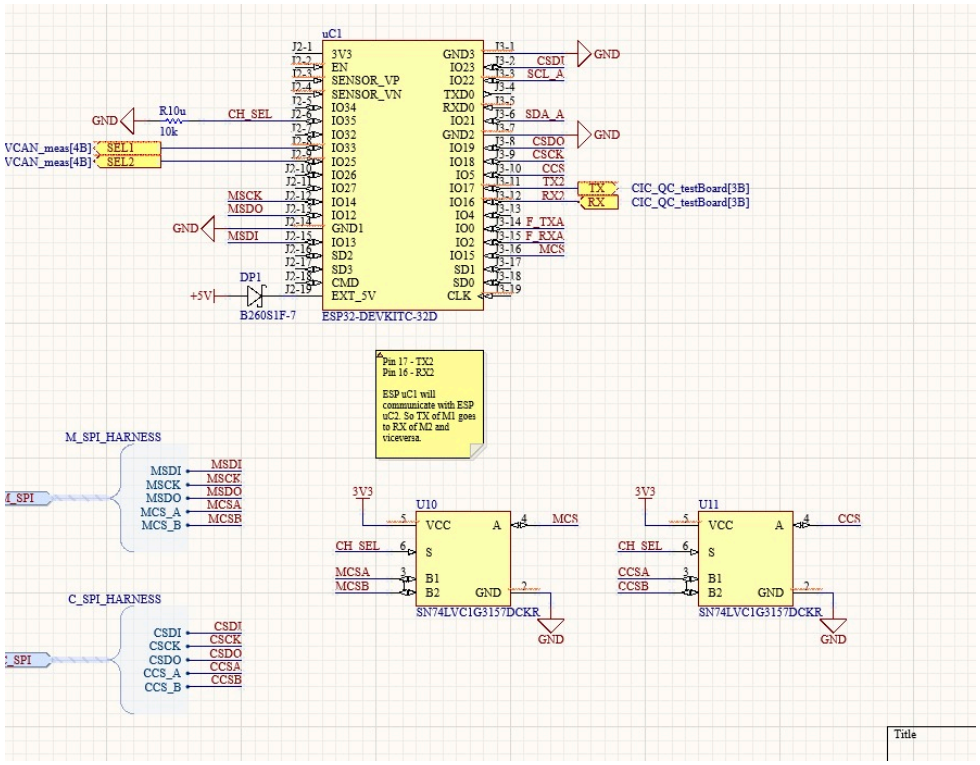
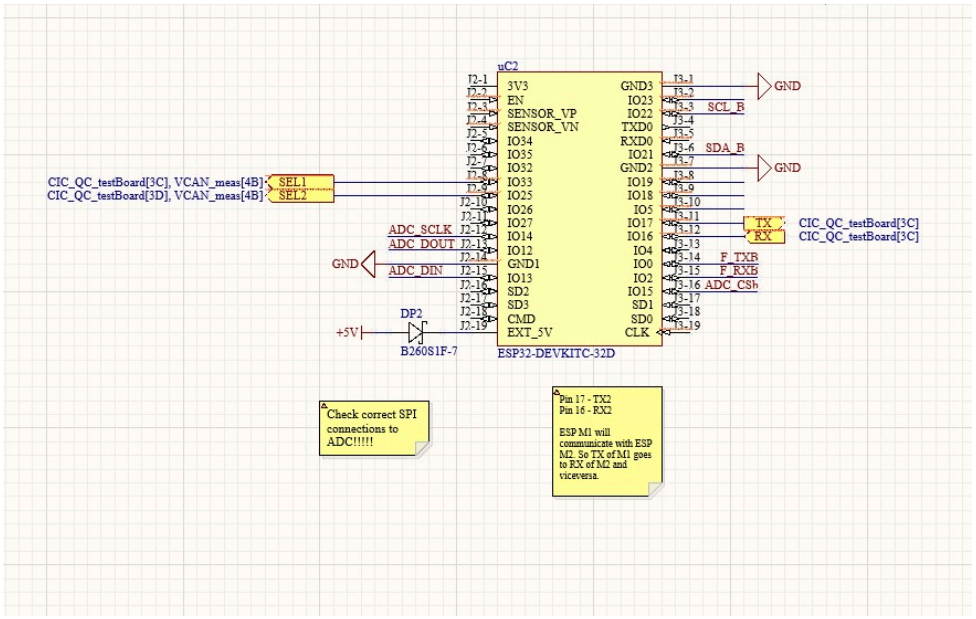
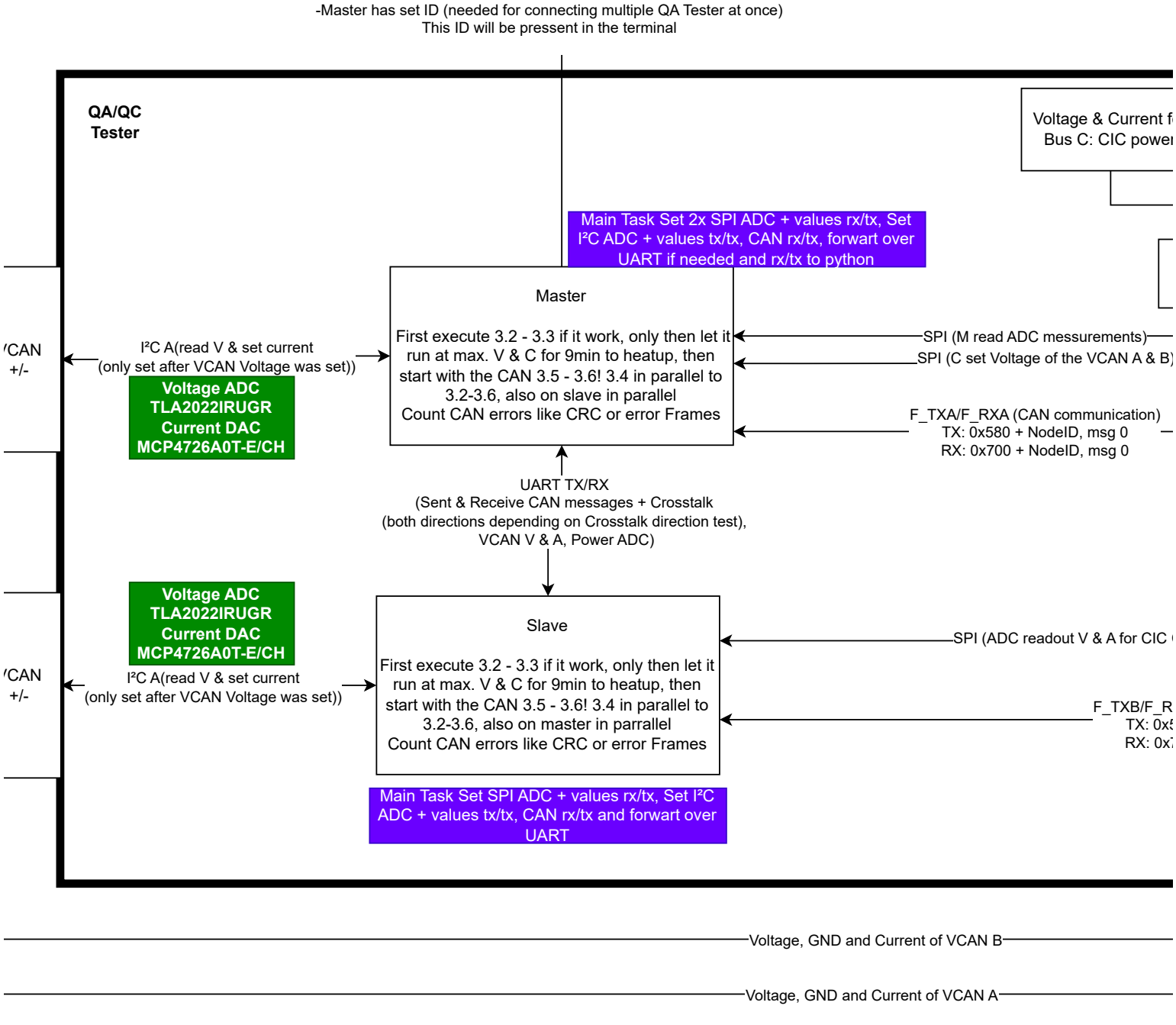

Mai task: Terminal input and output, Sent states to master, a csv, depending on 3.X, fill the CSV with the information
Menue + Options:
1. Enter Serial Number or use Barcode Scanner
After 1 Print SN, Name, Date, Time
2. All values OK? (Name, Date, Time, etc. Time and Date autimatical)
2.1 Editor menue
2.2 OK, then 3.
3.1 START ALL
3.2 Test Voltage CH: A| B| Both (Only A or B at the same time)
3.3 Test Current CH: A| B| Both (Only A or B at the same time and Voltage must be set before current)
3.4 Test Temperature CH: A| B| Both (Only A or B at the same time)
3.5 Test CAN CH: A| B| Both (for 1 min or 1000msg?)
3.6 Test Crosstalk CH: A aktive| B aktive (for 1 min or 1000msg?)
4. After 3.1,.2,....6: Display results and then Save & END + override file if already existing|| repeat 1.X, 2.X or 3.X? || NEW Test with differen t CIC connected?
Optional: Enter current load test (terminal: set current in a range + Ender, current is set; it changes after another commant + Enter; it ends on tpye END or end + Enter
Python script
PC
New CIC connected
UART
TXD0/RXD0
(Sent log of:
- CAN A & B of RX/TX
- VCAN V A & B
- VCAN V & A of CIC
- CIC ADC mesurements

TODO Hieraus ein flow diagram machen damit jeder nachvollziehen kann was passiert




uC1
ESP32-DEVKITC-32D
B260S1F-7
DP1
R10u
10k
CH_SEL
SEL1
SEL2
MSCK
MSDO
MSDI
CSDI
SCL_A
SDA_A
CSDO
CSCK
TX2
RX2
F_TXA
F_RXA
MCS
CIC_QC_testBoard[3B]
Pin 17 - TX2
Pin 16 - RX2
ESP uC1 will communicate with ESP uC2. So TX of M1 goes to RX of M2 and viceversa.
M_SPI_HARNESS
C_SPI_HARNESS
U10
U11
SN74LVC1G3157DCKR
MCSA
MCSB
CCS
CCSA
CCSB
Title


uC2
ESP32-DEVKITC-32D
B260S1F-7
DP2
SEL1
SEL2
CIC_QC_testBoard[3C], VCAN_meas[4B]
CIC_QC_testBoard[3D], VCAN_meas[4B]
ADC_SCLK
ADC_DOUT
ADC_DIN
SDA_B
SCL_B
F_TXB
F_RXB
ADC_CSb
CIC_QC_testBoard[3C]
Check correct SPI connections to ADC!!!!!
Pin 17 - TX2
Pin 16 - RX2
ESP M1 will communicate with ESP M2. So TX of M1 goes to RX of M2 and viceversa.


-Master has set ID (needed for connecting multiple QA Tester at once)
This ID will be pressent in the terminal
QA/QC
Tester
Voltage & Current f
Bus C: CIC powe
Main Task Set 2x SPI ADC + values rx/tx, Set I²C ADC + values tx/tx, CAN rx/tx, forwart over UART if needed and rx/tx to python
Master
First execute 3.2 - 3.3 if it work, only then let it run at max. V & C for 9min to heatup, then start with the CAN 3.5 - 3.6! 3.4 in parallel to 3.2-3.6, also on slave in parallel
Count CAN errors like CRC or error Frames
'CAN
+/-
I²C A(read V & set current
(only set after VCAN Voltage was set))
Voltage ADC
TLA2022IRUGR
Current DAC
MCP4726A0T-E/CH
SPI (M read ADC messurements)
SPI (C set Voltage of the VCAN A & B)
F_TXA/F_RXA (CAN communication)
TX: 0x580 + NodeID, msg 0
RX: 0x700 + NodeID, msg 0
UART TX/RX
(Sent & Receive CAN messages + Crosstalk
(both directions depending on Crosstalk direction test),
VCAN V & A, Power ADC)
Voltage ADC
TLA2022IRUGR
Current DAC
MCP4726A0T-E/CH
Slave
First execute 3.2 - 3.3 if it work, only then let it run at max. V & C for 9min to heatup, then start with the CAN 3.5 - 3.6! 3.4 in parallel to 3.2-3.6, also on master in parrallel
Count CAN errors like CRC or error Frames
'CAN
+/-
I²C A(read V & set current
(only set after VCAN Voltage was set))
SPI (ADC readout V & A for CIC
F_TXB/F_R
TX: 0x
RX: 0x
Main Task Set SPI ADC + values rx/tx, Set I²C ADC + values tx/tx, CAN rx/tx and forwart over UART
Voltage, GND and Current of VCAN B
Voltage, GND and Current of VCAN A

then USER switch all SWITCHES TO ON. Afterwords an automated
bit cobination testing (256 bits) happens to verify that only on the correct combination
1,25V;1,9V;2,0V;2,4V;2,6V;2,8V;3,5V;4,2V;4,6V

for
r

Voltage & Current for
Bus M: ADC of the CIC

ADC
ADC124S021CIMMX/NOPB

ADC 4x
channel

SPI M:
CS5523-ASZ

SPI C:
MCP23S08-E/SO

C & M)

XB (CAN communication)
580 + NodeID, msg 0
700 + NodeID, msg 0

CIC

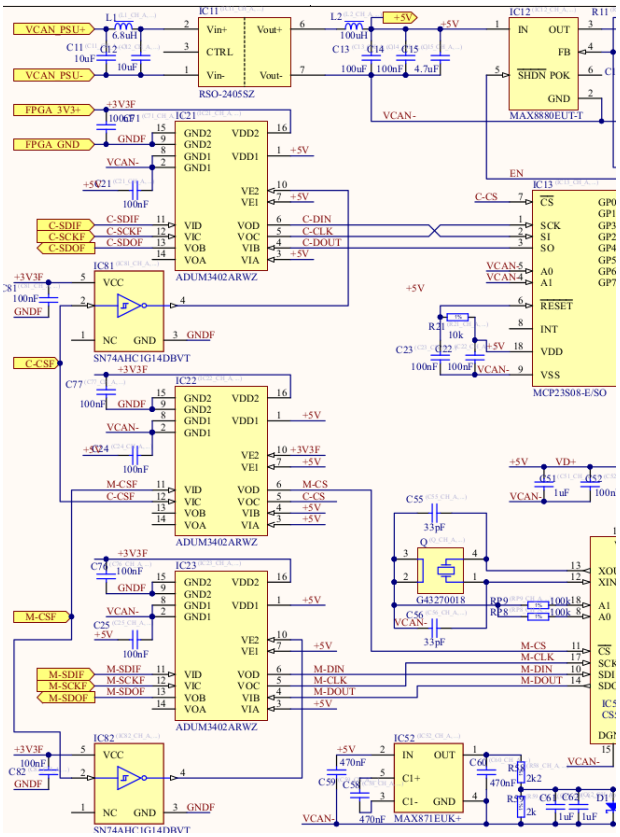
Switch Box

Level Shifter

Arduino Shield
CAN Receive: ID = 0x581, msg
CAN Sent: ID = 0x701, msg
Receive and Sent is on CAN an

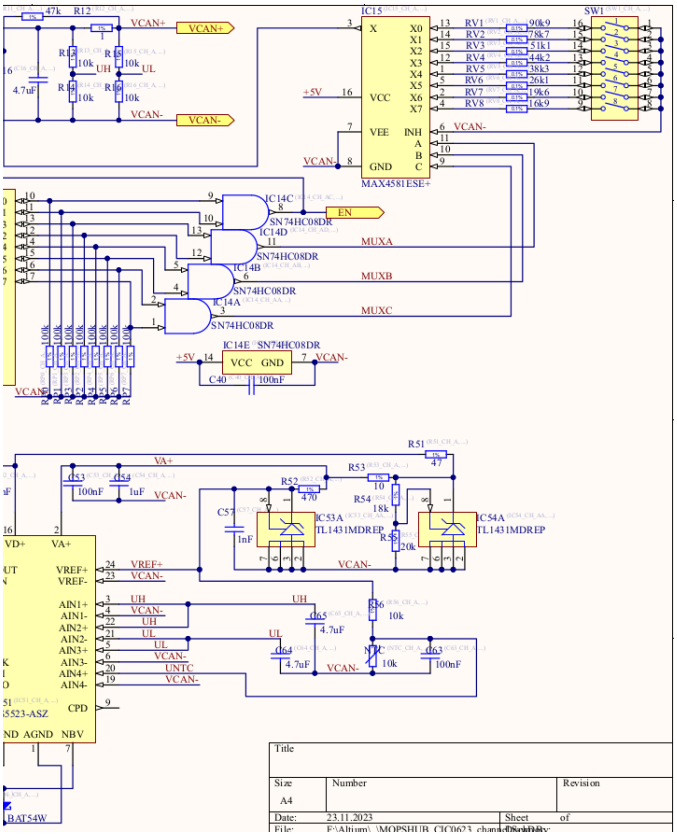
Level Shifter

Arduino Shield
CAN Receive: ID = 0x582, msg
CAN Sent: ID = 0x702, msg
Receive and Sent is on CAN an



NodeID = 1:
= 00, 00, 00, 00, 00, 00, 00, 00
= 00, 00, 00, 00, 00, 00, 00, 00
d for testing on Adruino UART

NodeID = 2:
= 00, 00, 00, 00, 00, 00, 00, 00
= 00, 00, 00, 00, 00, 00, 00, 00
d for testing on Adruino UART



Preface

The new version of the MOPSHUB-CIC has significant changes in controling the output voltage. Now all 8 Bits of the register chip MCP23S08 are used:

Version:	OLD CIC versions	NEW CIC version 0622
Set VCAN = 0 (Power disable)	Set Bit 0 = 0 (Bits 1 to 7 = dont care)	Set Bit 0 or Bit 1 (or both Bits) to 0
Set VCAN to the selected value (Power enable)	Set Bit 0 = 1 (Bits 1 to 7 = dont care)	Set Bit 0 AND Bit 1 to 1
Setting of VCAN voltage	By trimmpotentiometer	8 fixed values Defined by 8 SMD resistors selected by multiplexer (see tables below)

Bitpattern definition

Bit	7	6	5	4	3	2	1	0
function	Dual bit C for voltage selection (MSB)		Dual bit B for voltage selection		Dual bit A for voltage selection (LSB)		Dual bit P for POWER ENABLE	

C = Bit 7 AND Bit 6; B = Bit 5 AND Bit 4; A = Bit 3 AND Bit 2; P = Bit 1 AND Bit 0;

Logic table

C	**B**	**A**	**P**	**VCAN (Volt)**	**R_set (kOhm)**
X	X	X	0	0	(dont care)
Voltage setting resistor disabled by DIL switch			1	1,25	(infinity)
0	0	0	1	1,9	90,9
0	0	1	1	2,0	78,7
0	1	0	1	2,4	51,1
0	1	1	1	2,6	44,2
1	0	0	1	2,8	38,3
1	0	1	1	3,5	26,1
1	1	0	1	4,2	19,6
1	1	1	1	4,6	16,9

Note 1: Voltages are defined by fix resistors, the formula is:

VCAN = 1,257 V x (Rset + 47 kOhm) / Rset = 1,257 V x (1 + 47 kOhm / Rset)

Note 2: On the CIC card is (for each VCAN channel A and B) an 8-bit DIL-switch. Each of the 8 resistors Rset has in series a switch to disable it. Setting this switch to OFF changes Rset to „infinity“ and the corresponding VCAN to the minimum value of 1.257 V. So for security the maximum VCAN can be selected by the switch („by hardware“).

Note 3: The formular in Note 1 results from the used voltage regulator (MAX8880) with internal reference of typ. 1,257 V. The output voltage is feed back by a voltage divider with 47 kOhm (Tolerance 1%) from output to feedback input, and Rset from feedback input to GND. Rset are selected (switched) by an analog multiplexer (MAX4581).