STN11XX

Multiprotocol OBD-II to UART Interpreter

Table of Contents

1.0 Overview **3**
2.0 Feature Highlights **3**
3.0 Typical Applications **3**
4.0 Pinout **4**
5.0 Pin Descriptions **4**
6.0 Absolute Maximum Ratings **5**
7.0 Electric Characteristics **5**
8.0 Communicating with the STN11XX **6**
9.0 AT Commands **7**
9.1 Supported ELM327 Programmable Parameters 10
10.0 ST Commands **11**
10.1 General ST Commands 12
10.2 Device ID Commands 12
10.3 CAN Specific ST Commands 13
10.4 ISO Specific ST Commands 13
10.5 Filtering ST Commands 13
10.6 Monitoring ST Commands 14
11.0 OBD Requests **14**
12.0 OBD Message Filtering **14**
12.1 Non-CAN Protocols 14
12.2 CAN Protocols 15
13.0 CAN Message Reception **16**
14.0 Error Messages **18**
Appendix A: Revision History **19**
Appendix B: Contact Information **20**

1.0 Overview

On-Board Diagnostics, Second Generation (OBD-II) is a set of standards for implementing a computer based system to control emissions from vehicles. It was first introduced in the United States in 1994, and became a requirement on all 1996 and newer US vehicles. Other countries, including Canada, parts of the European Union, Japan, Australia, and Brazil adopted similar legislation. A large portion of the modern vehicle fleet supports OBD-II or one of its regional flavors.

Among other things, OBD-II requires that each compliant vehicle be equipped with a standard diagnostic connector (DLC) and describes a standard way of communicating with the vehicle's computer, also known as the ECU (Electronic Control Unit). A wealth of information can be obtained by tapping into the OBD bus, including the status of the malfunction indicator light (MIL), diagnostic trouble codes (DTCs), inspection and maintenance (I/M) information, freeze frames, VIN, hundreds of real-time parameters, and more.

The STN11XX is an OBD to UART interpreter that can be used to convert messages between any of the OBD-II protocols currently in use, and UART. It is fully compatible with the *de facto* industry standard ELM327 command set. Based on a 16-bit processor core, the STN11XX offers more features and better performance than any other ELM327 compatible IC.

2.0 Feature Highlights

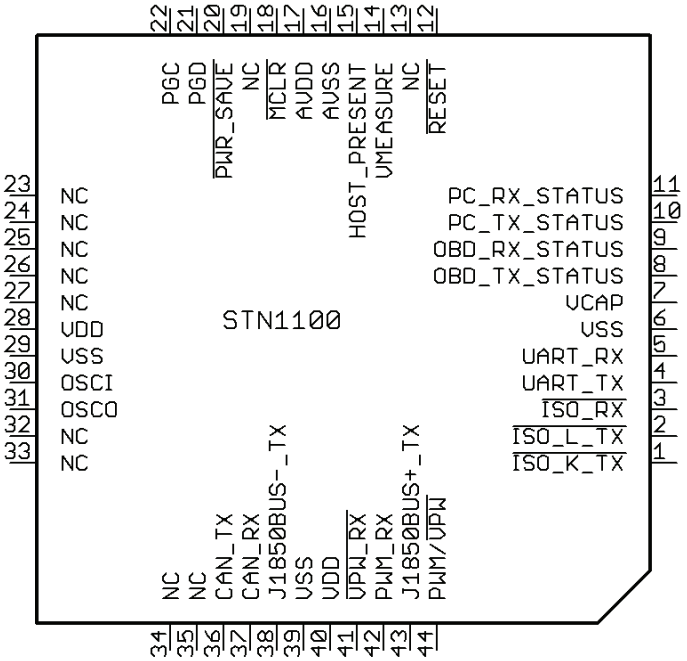
- Fully **compatible with the ELM327** AT command set
- **Extended** ST **command set**
- **UART interface** (baud rates from 38 bps to 10 Mbps[1])
- Secure **bootloader** for easy firmware updates
- Support for **all legislated OBD-II protocols:**
  - ISO 15765-4 (CAN)
  - ISO 14230-4 (Keyword Protocol 2000)
  - ISO 9141-2 (Asian, European, Chrysler vehicles)
  - J1850 VPW (GM vehicles)
  - J1850 PWM (Ford vehicles)
- Support for **non-legislated OBD protocols:**
  - ISO 15765
  - ISO 11898 (raw CAN)
- Superior **automatic protocol detection** algorithm
- **Large memory buffer**
- Voltage input for **battery monitoring**
- Available in **TQFP-44** and **QFN-44** packages

Note 1: Maximum theoretical baud rate. Actual maximum baud rate is application dependent and may be limited by driver hardware.

3.0 Typical Applications

- Academic projects
- Automotive diagnostic scan tools and code readers
- Digital dashboards
- Fleet management and tracking applications
- OBD data loggers

4.0 Pinout



5.0 Pin Descriptions

Pin #	Pin Name	Pin Type	Pin Description
1	$\overline{\text{ISO\_K\_TX}}$	O	Active low K-line transmit for ISO 9141/ISO 14230
2	$\overline{\text{ISO\_L\_TX}}$	O	Active low L-line transmit for ISO 9141/ISO 14230
3	$\overline{\text{ISO\_RX}}$	I	Active low receive for ISO 9141/ISO 14230
4	UART_TX	O	UART transmit
5	UART_RX	I	UART receive
6	VSS	P	Ground reference for logic and I/O pins
7	VCAP	P	CPU core voltage regulator filter capacitor connection
8	OBD_TX_STATUS	O	OBD transmit activity status pin
9	OBD_RX_STATUS	O	OBD receive activity status pin
10	PC_TX_STATUS	O	UART transmit activity status pin
11	PC_RX_STATUS	O	UART receive activity status pin
12	$\overline{\text{RESET}}$	I	Active low “reset to factory defaults” input

14	VMEASURE	A	Analog voltage input (see ATRV command description)
15	HOST_PRESENT	I	“Host present” input (used for power save functionality)
16	AVSS	P	Ground reference for VMEASURE input
17	AVDD	P	Analog positive supply
18	$\overline{\text{MCLR}}$	I	Active low “master clear” input
20	$\overline{\text{POWER\_SAVE}}$	O	Active low “power save” output (puts external circuitry into low power mode)
28	VDD	P	Positive supply for peripheral logic and I/O pins
29	VSS	P	Ground reference for logic and I/O pins
30	OSCI	I	Oscillator crystal input
31	OSCO	O	Oscillator crystal output
36	CAN_TX	O	CAN transmit
37	CAN_RX	I	CAN receive
38	J1850BUS-_TX	O	J1850 Bus- transmit
39	VSS	P	Ground reference for logic and I/O pins
40	VDD	P	Positive supply for peripheral logic and I/O pins
41	$\overline{\text{VPW\_RX}}$	I	Active low J1850 VPW receive
42	PWM_RX	I	J1850 PWM receive
43	J1850BUS+_TX	O	J1850 Bus+ transmit
44	PWM/ $\overline{\text{VPW}}$	O	J1850 PWM/VPW Bus+ voltage select

Legend

—	Unused pin	I	Input	P	Power
A	Analog input	O	Output		

6.0 Absolute Maximum Ratings

Reference Microchip PIC24HJ128GPX04 datasheet.

7.0 Electric Characteristics

Reference Microchip PIC24HJ128GPX04 datasheet.

8.0 Communicating with the STN11XX

The STN11XX uses a three-wire UART connection that is CMOS/TTL compatible. The UART settings are:

- 38400 baud (default)
- 8 data bits
- No parity bit
- One stop bit
- No handshaking

The baud rate is software-selectable (see STBR).

Note: *The UART Tx pin is configured as an open drain output and requires a 10 kΩ pull-up resistor. Maximum pull-up voltage is 5 volts.*

Once powered and connected, the STN11XX will display the welcome prompt:

```
ELM327 v1.3a

>
```

The STN11XX sends the '>' ("prompt") character, to signal that it is ready for more input. User software should always wait for the prompt before sending the next command.

There are three types of commands recognized by the STN11XX: **AT commands, ST commands,** and **OBD requests.**

The STN11XX is designed to fully emulate the ELM327 **AT command set** supported by many existing OBD software applications. AT commands begin with "AT" and are intended for the IC. They cause the STN11XX to carry out some action – change or display settings, perform a reset, and so on. A list of supported AT commands can be found in section 9.0.

In order to provide additional functionality while maintaining compatibility with the ELM327 command set, the STN11XX supports a parallel **ST command set**, described in section 10.0.

OBD requests are messages that are transmitted on the OBD bus. Only ASCII hexadecimal digits (0-9 and A-F) are allowed in OBD requests.

Only ASCII alpha characters, numbers, backspaces, and the carriage return are accepted on the UART, spaces are ignored. All commands must terminate with a carriage return (0x0D).

By default, responses from the STN11XX are terminated with a carriage return (0x0D). ATL1 command can be used to have the STN11XX append line feeds (0x0A) to the carriage returns.

Sending a single carriage return character repeats the last command.

9.0 AT Commands

This section lists the AT commands supported by the STN11XX. Every effort was made to maintain compatibility with legacy ELM327 software, and for most purposes, these commands work exactly as described in the ELM327 datasheet. Please refer to the "AT Commands" section of the ELM327 datasheet for the complete description of the AT command set.

Asterisk (*) marks default setting.

General Commands		
Command	**Description**	**Status**
<CR>	Repeat last command	supported
BRD hh	Try baud rate divisor hh	supported
BRT hh	Set baud rate timeout	supported
D	Set all settings to defaults	supported
E 0/1	Echo off/on*	supported
FE	Forget events	not applicable
I	Print ELM327 version ID string	supported [1]
L 0/1	Linefeeds off*/on	supported
M 0/1	Memory off/on*	supported
WS	Warm start	supported
Z	Reset device	supported
@1	Print manufacturer string	supported [2]
@2	Display device identifier	supported
@3	Store device identifier	supported

Programmable Parameter Commands		
Command	**Description**	**Status**
PP xx OFF	Disable PP xx	supported
PP FF OFF	All PPs off	supported
PP xx ON	Enable PP xx	supported
PP FF ON	All PPs on	supported
PP xx SV yy	For PP xx, set value to yy	supported
PPS	Print PP summary	supported

Voltage Reading Commands		
Command	**Description**	**Status**
CV dddd	Calibrate voltage to dd.dd volts	supported
RV	Read voltage	supported

OBD Commands		
Command	**Description**	**Status**
AL	Allow long (>7 byte) messages	supported
AR	Automatically receive	supported
AT 0/1/2	Adaptive timing off, auto1*, auto2	supported

BD	Buffer dump	not applicable
BI	Bypass initialization sequence	supported
DP	Describe current protocol	supported
DPN	Describe current protocol by number	supported
H 0/1	Headers off*/on	supported
MA	Monitor all	supported
MR hh	Monitor for receiver hh	supported
MT hh	Monitor for transmitter hh	supported
NL	Normal length messages* (7 bytes max)	supported
PC	Protocol close	supported
R 0/1	Responses off/on*	supported
RA hh	Set the receive address to hh	supported
S 0/1	Print spaces off/on*	supported
SH hhh	Set header to hhh	supported
SH hh hh hh	Set header to hh hh hh	supported
SP h	Set protocol to h and save it	supported [3]
SP Ah	Set protocol to h with auto search and save it	supported [3]
SR hh	Set receive address to hh	supported
ST hh	Set timeout to hh x 4 ms	supported
TP h	Try protocol h	supported [3]
TP Ah	Try protocol h with auto search	supported [3]

J1850 Specific Commands (protocols 1 and 2)		
Command	**Description**	**Status**
IFR 0/1/2	IFRs off, auto*, or on	supported
IFR H/S	IFR value from header* or source	supported

ISO Specific Commands (protocols 3 to 5)		
Command	**Description**	**Status**
IB 10	Set ISO baud rate to 10400*	supported
IB 96	Set ISO baud rate to 9600	supported
IIA hh	Set the ISO (slow) init address to hh	supported
KW	Display ISO key word	supported
KW 0/1	Key word checking off/on*	supported
SW hh	Set wakeup interval to hh x 20 ms	supported
WM [1-6 bytes]	Set wakeup wessage	supported

CAN Specific Commands		
Command	**Description**	**Status**
CAF 0/1	Automatic formatting off/on*	supported
CF hhh	Set ID filter to hhh	supported

CF hh hh hh hh	Set ID filter to hh hh hh hh	supported
CFC0/1	Flow control off/on*	supported
CM hhh	Set ID mask to hhh	supported
CM hh hh hh hh	Set ID mask to hh hh hh hh	supported
CP hh	Set CAN priority to hh (29 bit only)	supported
CRA hhh	Set CAN receive address to hhh	supported
CRA hh hh hh hh	Set CAN receive address to hh hh hh hh	supported
CS	Show CAN status counts	supported
D 0/1	Display of the DLC off*/on	supported
FC SM h	Flow control, set mode to h	supported
FC SH hhh	Flow control, set header to hhh	supported
FC SH hh hh hh hh	Flow control, set header to hh hh hh hh	supported
FC SD [1-5 bytes]	Flow control, set data to […]	supported
RTR	Send an RTR message	supported
V 0/1	Use of variable DLC off*/on	supported

J1939 CAN Specific Commands (protocols A to C)		
Command	**Description**	**Status**
DM1	Monitor for DM1 messages	not supported
JE	Use J1939 ELM data format*	not supported
JS	Use J1939 SAE data format	not supported
MP hh hh	Monitor for PGN 0hhhh	not supported
MP hh hh hh	Monitor for PGN hhhhhh	not supported

Note 1: ID string corresponds to the version of the ELM327 IC that STN11XX is designed to be compatible with (e.g., “ELM327 v1.3a”)

Note 2: Manufacturer string can be modified using STS@1 command.

Note 3: Current version of the STN11XX only supports protocols 1 through 9. However, to maintain compatibility, setting protocol number to any valid ELM327 protocol causes STN11XX to print “OK”. Attempts to send messages on unsupported protocols will result in NO DATA being returned by the scan tool.

9.1 Supported ELM327 Programmable Parameters

Programmable parameters are configuration values stored in non-volatile memory. Please refer to the "Programmable Parameters" section of the ELM327 datasheet for a full description of this functionality.

All programmable parameters can be turned off and reset to their default values by holding $\overline{\text{RESET}}$ input low for 5 seconds, until PC Rx LED starts flashing rapidly. After $\overline{\text{RESET}}$ input is released, device will set all factory defaults, and then perform ATZ reset.

Programmable Parameters				
PP	**Description**	**Values**	**Default**	**Type**
00	Perform ATMA after power up or reset	00 = ON FF = OFF	FF (OFF)	R
01	Printing of header bytes (ATH default setting)	00 = ON FF = OFF	FF (OFF)	D
02	Allow long messages (ATAL default setting)	00 = ON FF = OFF	FF (OFF)	D
03	NO DATA timeout time (ATST default setting) setting = value x 4.096 ms	00 to FF	32 (205 ms)	D
04	Default adaptive timing mode (ATAT setting)	00 to 02	01	D
06	OBD source (tester) address.	00 to FF	F1	R
07	Last protocol to try during automatic searches	01 to 0C	09	I
09	Character echo (ATE default setting)	00 = ON FF = OFF	00 (ON)	R
0A	Line feed character	00 to FF	0A	R
0C	Default UART baud rate divisor setting = 4000000 / value	02, 04, 06 to FF	68 (38.4 kbps)	P
0D	Carriage return character	00 to FF	0D	R
10	J1850 voltage settling time setting = value x 4.096 ms	00 to FF	0D (53 ms)	I
13	Auto search time delay between protocols 1 & 2 setting = value x 4.096 ms	00 to FF	32 (205 ms)	I
16	Default ISO baud rate (ATIB default setting)	00 = 96 FF = 10	FF (10.4 kbps)	R
17	ISO wakeup message rate (ATSW default setting) setting = value x 20.48 ms	00 to FF	92 (2.99 sec)	D
18	Auto search time delay between protocols 4 & 5 setting = value x 4.096 ms	00 to FF	00 (no delay)	I
24	CAN auto formatting (ATCAF default setting)	00 = ON FF = OFF	00 (ON)	D
25	CAN auto flow control (ATCFC default setting)	00 = ON FF = OFF	00 (ON)	D
26	CAN filler byte (used to pad out messages)	00 to FF	00	D
29	Printing of CAN data length (DLC) when printing header bytes (ATD0/1 default setting)	00 = ON FF = OFF	FF (OFF)	D

10.0 ST Commands

ST commands are designed to provide extended functionality, without breaking compatibility with the ELM327 AT command set. Both command sets are available simultaneously.

General	
Command	**Description**
BR *baud*	Switch UART baud rate
BRT *ms*	Set UART baud rate switch timeout
S@1 *ascii*	Set AT@1 device description string

Device ID	
Command	**Description**
DI	Print device hardware ID string (e.g., “OBDLink r1.7”)
I	Print firmware ID string (e.g., “STN1100 v1.2.3”)
MFR	Print device manufacturer ID string
SN	Print device serial number

CAN Specific	
Command	**Description**
CAFCP *ttt* ,*rrr*	Add flow control 11-bit ID pair
CCFCP	Clear all Flow Control 11-bit ID Pairs

ISO Specific	
Command	**Description**
IAT 0/1	Turn adaptive maximum interbyte timing (P_1 max) off/on*
IBR *baud*	Set ISO baud rate
IMCS 0/1	Turn ISO manual checksum off*/on
IP1X *ms*	Set maximum interbyte time for receiving messages (P_1 max)
IP4 *ms*	Set interbyte time for transmitting messages (P_4)

Filtering	
Command	**Description**
FAP *[pattern]* , *[mask]*	Add pass filter
FAB *[pattern]* , *[mask]*	Add block filter
FAFC *[pattern]* , *[mask]*	Add flow control filter
FCP	Clear all Pass filters
FCB	Clear all Block filters
FCFC	Clear all Flow Control filters

Monitoring	
Command	**Description**
M	Monitor OBD bus using current filters
MA	Monitor all messages on OBD bus

10.1 General ST Commands

BR *baud*

Switch UART baud rate. The STBR command operates identically to ATBRD command, with the following exceptions:

- Baud rate is specified as a decimal number in baud
- Returns '?' if the specified baud rate cannot be generated with 3% or better accuracy
- The ID string returned is the STI string

Examples:

```
STBR 300        switch baud rate to 300 bps
STBR 115200     switch baud rate to 115.2 kbps
STBR 2000000    switch baud rate to 2 Mbps
```

See ELM327 datasheet for more information about baud rate switching algorithm.

BRT *ms*

Set UART baud rate switch timeout for ATBRD and STBR commands. The STBRT command sets the same timeout as the ATBRT command, except that the timeout is specified as a decimal value in milliseconds and the maximum timeout is 65535 ms (65.5 seconds).

S@1 *ascii*

Set the device description string returned by AT@1 command. Accepts printable ASCII characters (0x20 to 0x7E). Leading and trailing spaces will be ignored.

10.2 Device ID Commands

STN11XX supports a number of commands which can be used to identify the device, get its unique serial number, and print the firmware and hardware versions.

DI

Print device hardware ID string, in this format:

<device_name> rX.Y

Table below lists device names for the devices currently in production, as well as devices still in development:

Device ID	Device Name
1000	OBDLink CI
1100	OBDLink
1101*	OBDLink S
1102*	microOBD
1110*	*one-time programmable*

***** *future device*

X.Y is the device hardware revision number.

Example: `OBDLink r1.7`

I

Prints firmware ID string, in this format:

STN<device_id> vX.Y.Z

X.Y.Z is the firmware version number.

Example: `STN1101 v1.1.0`

MFR

Print the device manufacturer ID string. On STN1110, this command returns "Generic" by default.

SN

Print the device serial number. Serial numbers for all devices are 12 digits long, and begin with the device ID, making each serial number unique across all STN11XX devices:

<device_id><serial_number>

Example: `110012345678`

The serial number is programmed at the factory and cannot be changed.

10.3 CAN Specific ST Commands

CAFCP *ttt*, *rrr*

Add a flow control 11-bit CAN ID pair. Takes two three-digit parameters: *ttt* is transmitter ID, and *rrr* is receiver ID. For example, STCAFCP 7E0,7E8.

CCFCP

Clear all flow control 11-bit ID pairs.

10.4 ISO Specific ST Commands

IAT 0/1

Turn ISO adaptive P_1 max timing off/on*. When this mode is on, maximum interbyte time (P_1 max) for ISO 9141 messages is adaptively reduced to allow communication with some ECUs that do not comply with the minimum intermessage time (P_2 min) specified in ISO 9141-2 standard. It is on by default.

IBR *baud*

Set ISO baud rate. Takes a decimal parameter, expressed in bits per second (bps). Supported baud rates are 612 to 65535 bps (65.5 kbps).

IMCS 0/1

Turn ISO manual checksum off*/on. When this setting is on, STN11XX will not automatically append checksum byte for transmitted messages, or verify checksum for received messages. Additionally, for KWP2000 protocols (4 and 5), minimum allowed OBD request length is increased to 2 bytes (one data byte and checksum).

IP1X *ms*

Set maximum interbyte time for receiving ISO messages (P_1 max). Takes a decimal parameter in milliseconds. Default is 20 ms.

IP4 *ms*

Set interbyte time for transmitting ISO messages (P_4). Takes a decimal parameter in milliseconds. Default is 5 ms.

10.5 Filtering ST Commands

Each of the Add Filter commands dynamically allocates a block of RAM to store the filter, and can return OUT OF MEMORY error if there is not enough memory to add the filter. If this occurs, OBD requests may start generating the OUT OF MEMORY errors because the OBD memory buffer is located in the same RAM.

FAP *[pattern]*, *[mask]*

Add a pass filter. Takes two parameters: pattern and mask. Pattern and mask can be any length from 0 to 5 bytes (0 to 10 ASCII characters), but both have to be the same length. The messages are matched MSB first, up to the filter length. Messages shorter than the filter length, will not match that filter.

If an odd number of ASCII characters is specified, a leading 0 will be added to the first byte. In other words,

```
STFAP 7E8,7FF
..is the same as
STFAP 07E8,07FF
```

For 29-bit CAN, the first four bytes are CAN ID; for 11-bit CAN, the first two bytes are CAN ID.

The first 3 bits for 29-bit CAN or the first 5 bits for 11-bit CAN should be don't care (0s in mask) and/or 0s in pattern.

FAB *[pattern]*, *[mask]*

Add block filter. Same syntax as STFAP.

FAFC *[pattern]*, *[mask]*

Add flow control filter. Same syntax as STFAP.

FCP

Clear all pass filters.

FCB

Clear all block filters.

FCFC

Clear all flow control filters.

10.6 Monitoring ST Commands

M

Monitor OBD bus using current filters.

MA

Monitor all messages on OBD bus. For CAN protocols, all messages will be treated as ISO 15765. To monitor raw CAN messages, use STM command.

11.0 OBD Requests

The STN11XX uses the same format for OBD requests as the ELM327. Please refer to the "OBD Commands" section of the ELM327 datasheet for information.

See the following standards for more information about legislated On-Board Diagnostics:

SAE J1979: E/E Diagnostic Test Modes. This document describes data reporting requirements of On-Board Diagnostic regulations in the United States and Europe, and any other region that may adopt similar requirements in the future. The ISO equivalent of this standard is ISO 15031-5.

SAE J2190: Enhanced E/E Diagnostic Test Modes. This document describes the implementation of Enhanced Diagnostic Test Modes, which are intended to supplement the legislated Diagnostic Test Modes defined in SAE J1979 standard. Modes are defined for access to emission related test data beyond what is included in SAE J1979, and for non-emission related data.

SAE J2178: Class B Data Communication Network Messages. This document describes the information contained in the header and data fields of non-diagnostic messages for automotive serial communications based on SAE J1850 Class B networks.

12.0 OBD Message Filtering

STN11XX supports pass, block, and flow control filters. Their operation is backwards compatible with the ELM327, however STN11XX filtering scheme is much more powerful and flexible. It allows the user to set up multiple filters and fine tune them to receive only those messages that are of interest to the user.

12.1 Non-CAN Protocols

Non-CAN protocols (see ATSP, protocols 1 through 5) do not use flow control filters (refer to Figure 1). When a message comes from the OBD bus, it is compared to the pass filters. If the message does not match one of the filters, it is discarded. Otherwise, the message is compared to the block filters. If there is a match, the message is discarded. Finally, if the message goes through both the pass and block filters, it is transmitted on the UART.

In **automatic filtering mode,** pass filters are automatically set based on the currently set message header. Table below lists the filters set up from the default headers:

Protocol(s)	Filter (pattern, mask)
J1850 VPW/PWM ISO 9141-2	006B00,14FF00
ISO 14230-4	80F100,C0FF00

While in the automatic filtering mode, anytime the message header is changed, either by the user (ATSH command) or because of a protocol change, the pass filter gets updated.

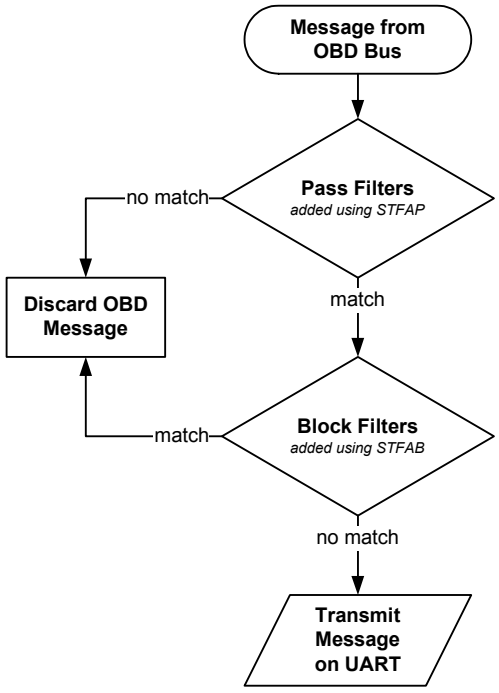


Figure 1 – Message Filtering: Non-CAN Protocols

As soon as the user clears the pass filters, or adds a pass filter, automatic filtering mode is switched off. Issue **ATAR** to clear all custom filters, set up default filters, and turn on the automatic filtering mode.

Some commands temporarily alter the contents of the pass filters.

For example, while the **ATMA** or **STMA** commands are active, they temporarily delete any previously added pass or block filters, and set up one “pass all” filter. Upon termination of the command, the “pass all” filter is removed, and the old pass/block filters are restored.

ATMR and **ATMT** commands behave the same way, except that instead of setting a “pass all” filter, they set up a filter to accept messages based on the address of the receive (or transmit) node passed as the parameter.

STM command uses all filters “as-set”: it does not modify them in any way.

ATSR turns off the automatic filtering mode, and sets up a pass filter to accept messages sent to the receive address provided as the parameter to ATSR.

In order to directly manipulate the filters, use the filtering ST commands described in section 10.5.

12.2 CAN Protocols

This section describes how message filtering works with CAN protocols (see ATSP, protocols 6 through 9).

When a CAN frame comes in from the network, it must first go through the CAN hardware filter. If there is no match, the frame is discarded. If there is a match, the frame is compared against the flow control filters to determine whether it is an ISO 15765 or an ISO 11898 (“raw”) CAN frame.

ISO 11898 frames are compared to the pass filters. If there is no match, the frame is discarded. Otherwise, the frame is compared to the block filters, and if there is no match, it is transmitted on the UART.

ISO 15765 frames bypass the pass filters. As long as the comparison with the block filters results in a “no match”, the frame is transmitted on the UART.

In **automatic filtering mode,** flow control filters are automatically set based on the currently set message header. Table below lists the filters set up from the default CAN headers:

CAN ID type	Filter (pattern, mask)
11-bit	7E8,7F8
29-bit	18DAF100,1FFFFF00

While in the automatic filtering mode, anytime the user changes the headers using the ATSH command, or by switching from 11-bit to 29-bit CAN IDs, the flow control filter gets updated.

Automatic filtering mode is switched off when the user clears the flow control filters, adds a flow control filter, or sets the CAN hardware filter. To clear all custom filters, and set up default filters, issue the **ATAR** command.

The **ATMA** command sets the flow control, pass, and block filters for “pass all, block none” operation. When the command terminates, the old filters are restored.

The **STMA** command works the same way as ATMA, except that it also sets the CAN hardware filter for “pass all” operation. Upon termination, the old CAN hardware filter is restored.

ATMR and **ATMT** commands behave the same way, except that instead of setting a “pass all” filter,

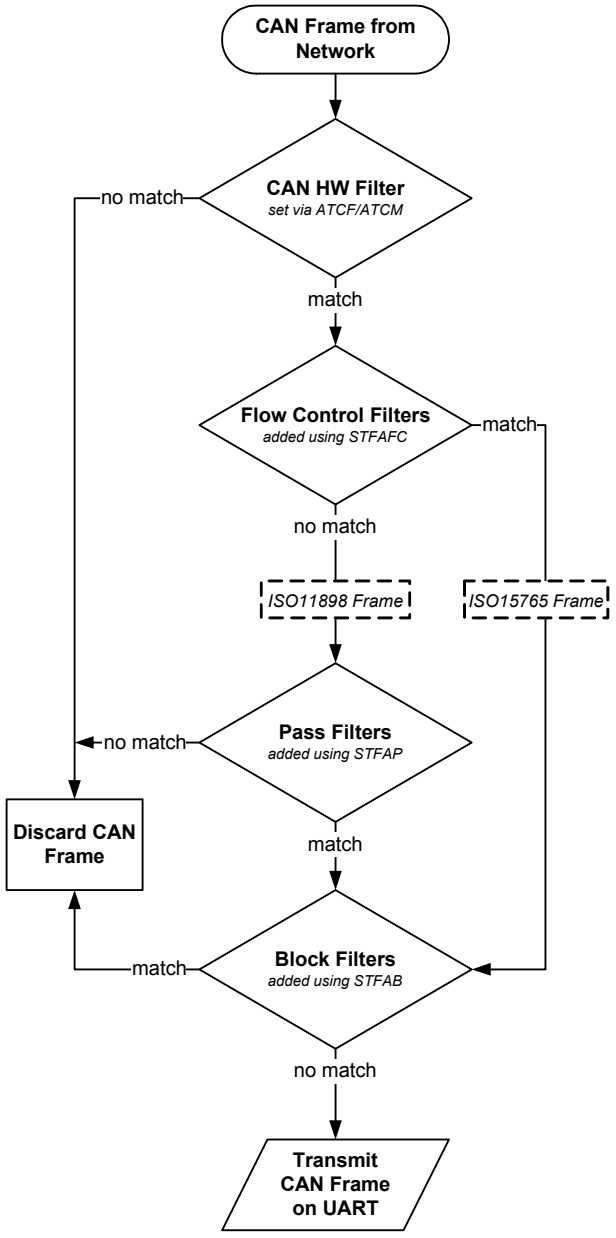


Figure 2 – Message Filtering: CAN Protocols

they set up a filter to accept messages based on the address of the receive (or transmit) node passed as the parameter.

STM command uses the filters "as-set": it does not modify them in any way.

ATSR turns off the automatic filtering mode, and sets up a pass filter to accept messages sent to the receive address provided as the parameter to ATSR.

In order to directly manipulate the filters, use the filtering ST commands described in section 10.5.

13.0 CAN Message Reception

For most users, CAN message reception works "out of the box", as configured by default. However, for those users who wish to take full advantage of the STN11XX's CAN architecture, it is important to understand what goes on behind the scenes.

You will notice that the flowchart in Figure 3 is

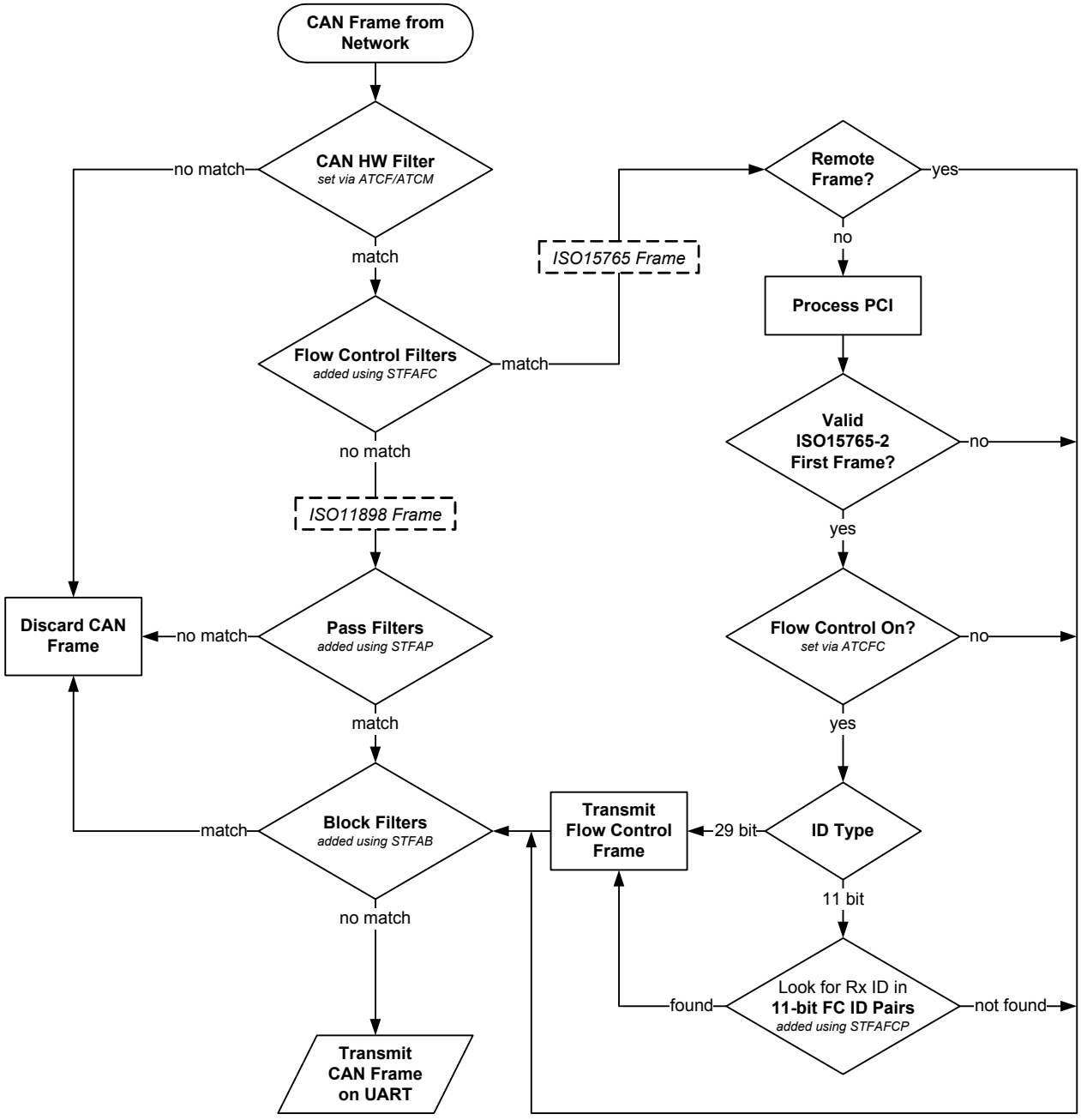


Figure 3 – CAN Message Reception

simply a more detailed version of the flowchart from Figure 2. Therefore, in this section we will omit the left half of the flowchart, and describe what happens when the incoming CAN frame is identified as an ISO 15765 CAN frame.

If the RTR bit is set, the frame is determined to be a **remote frame.** As long as it is not discarded by the block filters, it gets sent over UART.

If the frame is not a remote frame, additional processing takes place. The protocol control information (PCI) byte is processed to determine whether it is a valid ISO 15765-2 frame, and what type of frame it is (single, first, consecutive, or flow control).

If the frame is not a valid ISO 15765-2 first frame, or if flow control is off, it is passed to the block filters.

If the frame is a valid ISO 15765-2 **first frame,** and **flow control** is on, what happens next is determined by the ID type.

A **29-bit** frame ID contains the address of the transmitter, therefore a flow control frame is transmitted for every ISO 15765-2 first frame, before the frame is passed to the block filters.

An **11-bit** frame is first compared to the list of flow control 11-bit ID pairs. If a matching Rx ID is found, a flow control frame is transmitted on the CAN bus using the corresponding Tx ID. Otherwise, no flow control frame is sent, and the received frame is passed directly to the block filters.

By default, when an 11-bit CAN protocol is selected, STN11XX defines the following flow control 11-bit ID pairs:

Tx ID	Rx ID
7E0	7E8
7E1	7E9
7E2	7EA
7E3	7EB
7E4	7EC
7E5	7ED
7E6	7EE
7E7	7EF

Note that when adding custom flow control filters for 11-bit CAN messages using the STFAFC command, it is important to add corresponding flow control 11-bit ID pair (STFAFCP command) if the user wants to have the flow control frames be sent.

14.0 Error Messages

When hardware problems or transmission errors are encountered, the STN11XX will display an error message.

OUT OF MEMORY

This error message is displayed when there is not enough available RAM to complete the requested operation. For example, this may happen if too many filters are set.

For a list of other possible error messages and their descriptions, please refer to the “Error Messages” section of the ELM327 datasheet.

Appendix A: Revision History

Revision B (December 4, 2009)

This revision includes minor typographical and formatting changes throughout the datasheet text.

Global changes include:

- Changed all instances of STN1100 to STN11XX

The following sections were added:

- 12.0 OBD Message Filtering
- 13.0 CAN Message Reception

The major changes are referenced by their respective section in the following table.

Section Name	Update Description
5.0 Pin Descriptions	Removed descriptions for NC pins. Added/updated pin descriptions.
9.0 AT Commands	Added support for the following commands: • AT@3 • ATIB 96 Described the changed functionality of AT@1 and AT@2. Added description for “reset to factory defaults” functionality (**Section 9.1**). Added programmable parameters 0C and 16 (**Section 9.1**).
10.0 ST Commands	Added the following commands: • STBR, STBRT • STS@1 • STDI, STMFR, STSN • STIAT, STIBR, STIMCS, STIP1X, STIP4 • STM, STMA

Revision A (September 11, 2009)

Initial release of this document.

Appendix B: Contact Information

ScanTool.net, LLC
1819 W Rose Garden Ln Ste 3
Phoenix, AZ 85027

Email: sales@scantool.net
Web: www.scantool.net