NA RNS-E Firmware Update Release Notes

Contents

1. Overview

Presented on this CD is an unofficial firmware release for the RNS-E “Audi Navigation Plus” navigation system. This release introduces a number of useful new features as well as fixing many outstanding bugs.

The firmware has been tested on both major hardware versions of the RNS-E. However, users are reminded that this software is provided “as-is” and that they install modified firmware entirely at their own risk. Also that as we are based in Europe, we are not able to perform “on the road” testing of any NA firmware.

In some rare situations, the RNS-E may request the 4-digit security code PIN following an update or reboot. We advise you ensure you have your PIN to hand before performing a firmware update.

Support and feedback rnse@pcbbc.co.uk

Check for future firmware updates and enhancements rnse.pcbbc.co.uk

If you found this firmware useful and wish to contribute paypal.me

2. Release History

September 2020 – Beech Release
First release for NA units

3. Identifying RNS-E Versions

Audi have a number of navigation systems, but we only support the RNS-E with our firmware. Please don't ask us about their other systems. They are based on a completely different hardware platform which we know nothing about.

An RNS-E has a fold down screen which reveals the DVD drive and 2 SDCard slots. Additionally, it has the text "Audi Navigation Plus" printed on the bottom of the bezel (except Seat versions). If your navigation system does not have these features, then it's NOT an RNS-E!

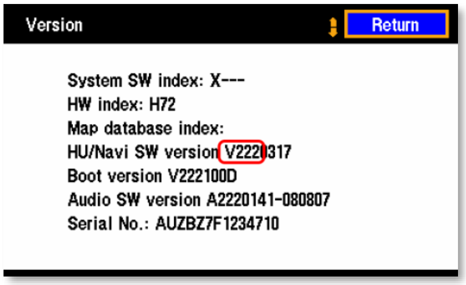
There were two different hardware models produced, which are usually referred to by the last 3 digits of their part numbers, either 192, or 193.

Here's how to tell the two models apart. You can check in the Engineering Menu “General → Version” screen for the HU/Navi SW version if you are unsure.

To access the Engineering Mode see Enabling Features via the Engineering Menus

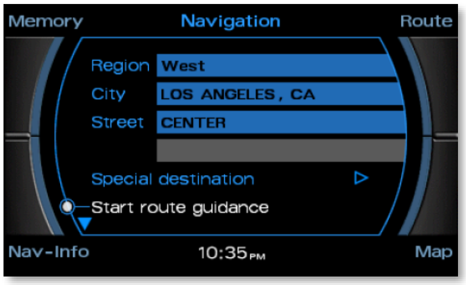
192 RNS-E (up until late 2009)

- Maximum 2 x 4GB SD cards (does **NOT** support SDHC cards)
- 400 x 240 LCD screen
- CD/TV button



System SW index: X---
HW index: H72
Map database index:
HU/Navi SW version V2220317
Boot version V222100D
Audio SW version A2220141-080807
Serial No.: AUZBZ7F1234710

V222 HU/Navi SW = 192 unit

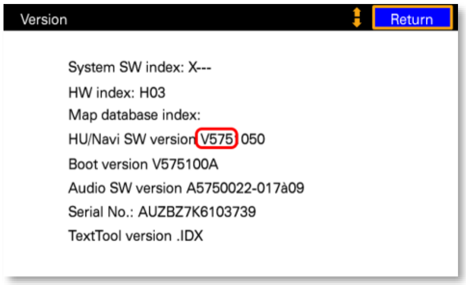


Region West
City LOS ANGELES, CA
Street CENTER

192 = Solid text box and frame border

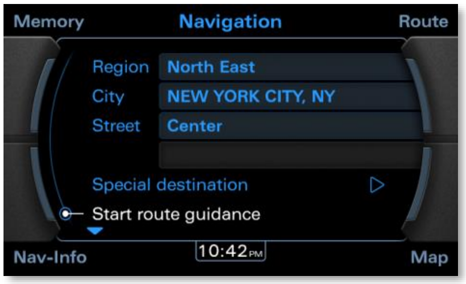
193PU RNS-E (from late 2009 on)

- Maximum 2 x 32GB SDHC cards
- 800 x 480 hi-res LED screen
- MEDIA button



System SW index: X---
HW index: H03
Map database index:
HU/Navi SW version V575 050
Boot version V575100A
Audio SW version A5750022-017à09
Serial No.: AUZBZ7K6103739
TextTool version .IDX

V575 HU/Navi SW = 193 PU unit



Region North East
City NEW YORK CITY, NY
Street Center

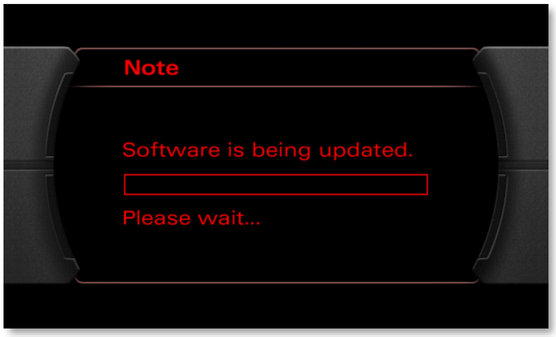
193 PU = Graduated text box and border

4. Installing RNS-E Firmware

To update your RNS-E…

- Burn this ISO image to a CD-R or CD-RW
- Remove your navigation DVD from the RNS-E and insert the CD
- After a few minutes the firmware update will start automatically
- Wait for the update to complete
  Note: Do ***not*** turn off the ignition while the firmware update is in progress
- Once the firmware update has completed the RNS-E will reboot
- Remove the CD from the DVD slot and replace with your regular navigation media

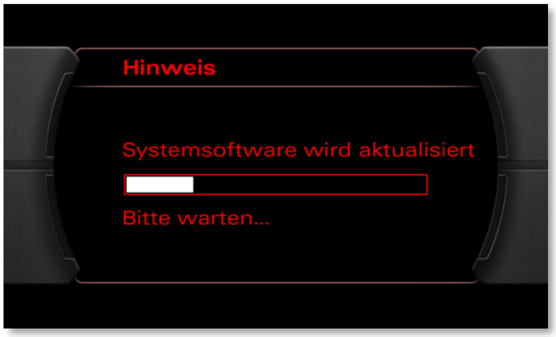
📋 Note: DVD-R or DVD-RW media are ***not*** required for firmware updates. Please use a CD.



Note

Software is being updated.

Please wait...

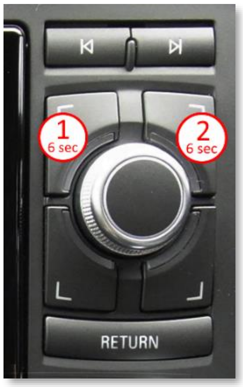


Hinweis

Systemsoftware wird aktualisiert

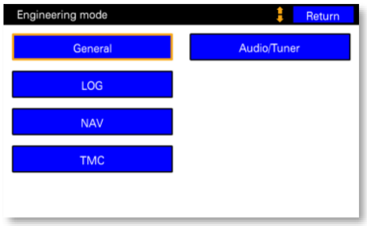
Bitte warten...

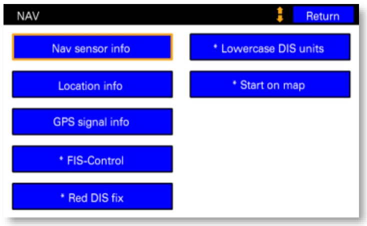
5. Enabling Features via the Engineering Menus



Additional features of this firmware are activated via the RNS-E Engineering Menu. To access the Engineering Menu:

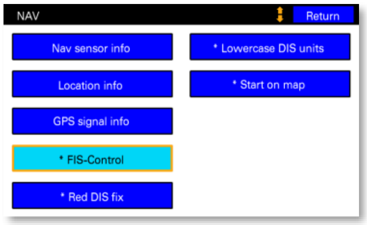
- Press NAV
- Press RETURN (until the Destination Entry screen is displayed)
- Press and hold the TOP LEFT quadrant key for at least 6 seconds, then release
- Press and hold the TOP RIGHT quadrant key until the Engineering Mode menu appears





Toggles for new features can be found under the Engineering Mode's "NAV" menu. Within each menu configurable features are prefixed with an asterisk.

Disabled features are shown with a dark blue background; enabled features with a light blue/cyan background. To enable or disable an option scroll to it using the main jog control and press select.



Under NAV menu you can find the options:

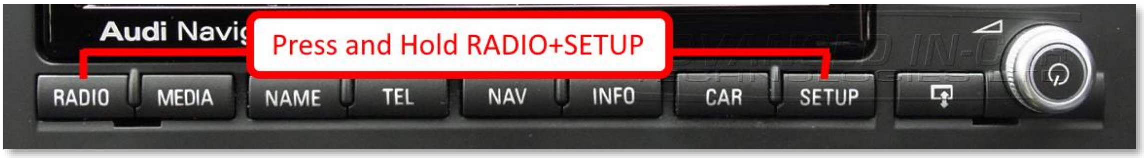
- FIS-Control – enables FIS-Control support
- Red DIS fix – enables a fix for the Red-DIS cluster bug
- Lowercase DIS units – displays KM/MI and AM/PM in lowercase for DIS directions
- Start on map – displays the map screen, in place of destination entry, on start up

See the New Features section for a full description of each feature.

📋 You can access items in the right column by scrolling to the top, and then back down
📋 To exit Engineering Mode, press RETURN until the RNS-E returns to the normal menus

6. Rebooting the RNS-E

To activate some features, it is necessary to reboot the RNS-E. To perform an immediate reboot press and hold the RADIO and SETUP buttons simultaneously for about 6 seconds until the unit reboots.



Alternatively, the selected feature will become active next time you use the car (after locking and a period of inactivity has expired such that the RNS-E fully powers down).

7. Reverting to an Earlier Firmware

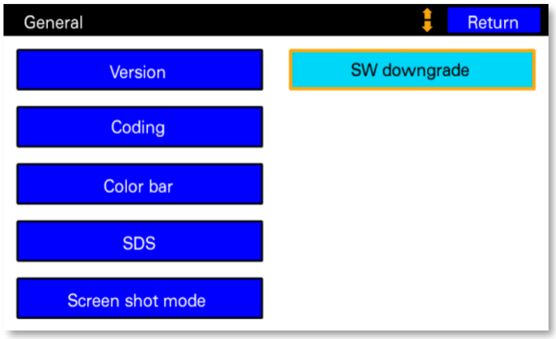
We hope you are happy with your new firmware. However, if you want to revert to the official firmware then you can do so at any time.

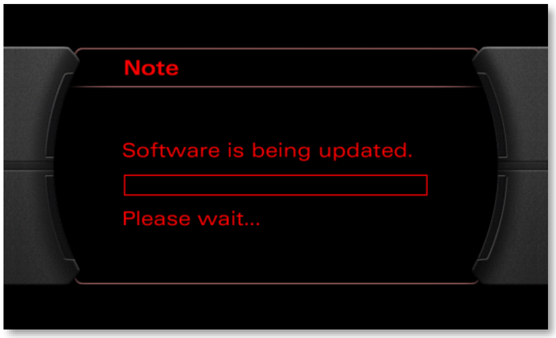
You can find an original Audi Firmware on any RNS-E Navigation DVD. Any disk from the 2016 release (8P0 060 884 CF) and onwards will contain the latest firmware versions for both 192 and 193 units.

Alternatively, you can download a CD copy from our downloads page.

All of our firmware releases contain an option in the Engineering Menus to perform a downgrade. Otherwise you can use VCDS and set adaption channel 97 to 1.

- Make sure you have the latest navigation DVD (Any DVD after 2016)
- If not, you should download a copy of the 230/260 firmware and burn to CD
- Remove the navigation DVD from the RNS-E
- Enter the RNS-E Engineering Mode
- Select option: “General -> Firmware Downgrade”
- Press RETURN until you leave Engineering Mode
- Insert the DVD or CD containing the older firmware
- Wait for the firmware downgrade to complete




Note

Software is being updated.

Please wait...

8. Release Notes – SW 0270 / SW 0300

NA 192 units - SW 0270
NA 193 PU units - SW 0300

8.1 New Features

8.1.1 Support for FIS-Control

When FIS-Control is installed on certain vehicles it is necessary for FIS-Control to share the cluster's CANBUS communications channel with the navigation system. This leads to a conflict for access to the cluster display if FIS-Control is used in conjunction with an RNS-E.

This firmware introduces the ability to switch the RNS-E to use an alternate CANBUS communication channel, and thus resolve this conflict. FIS-Control can then be started in "CAN Bridge" mode where it will re-route messages between the RNS-E and the cluster.

To enable/disable FIS-Control support, see Enabling Features via the Engineering Menus
Following activation, a reboot is required, see Rebooting the RNS-E

📋 In FIS-Control enable the "CAN Bridge" User Mod:
Settings → User Mods → CAN Bridge
Save settings and restart the ignition

📋 To start FIS-Control in RNS-E CAN Bridge Mode, press the UP stalk button when turning on the ignition

8.1.2 Show Map on Cold Boot

By default, the Navigation Sub-System displays the Destination Entry screen following a cold boot.

This firmware includes an option to have the unit automatically switch to the Map screen instead.

To enable/disable showing the map on boot, see Enabling Features via the Engineering Menus
The new setting will take effect the next time the RNS-E is cold booted.

📋 When the RNS-E reboots it displays the last selected sub-system, RADIO, MEDIA, NAV etc.
📋 To have the map display at start, leave the RNS-E in NAV mode when leaving the car
📋 To enter a destination, press RETURN or the Bottom Right Quadrant Key from the Map View

8.1.3 Reverse Gear over CAN

When a basic reversing camera is installed the camera can be selected via a 12v signal on the RNS-E C2 pin. Unfortunately, some later 193PU models of RNS-E lack the physical circuitry that implemented the C2 function. For these units it was not possible to have the camera input auto select on reverse.

This firmware introduces the ability for either RFSL C2 (if present) or the CANBUS reverse gear message to select the basic reversing camera.

📋 Not all vehicles support reverse over CANBUS – for example it is not present on earlier vehicles such as the TT 8N.
📋 You can check if your RNS-E is correctly detecting reverse via the Engineering Menu:
NAV → Nav senor info → Vehicle Signal REV OFF/ON

8.1.4 CAN Reverse Gear for A6 Mode (193PU only)

As discussed above the A6 uses only the dedicated RFSL C2 pin as reverse is not present on CAN. This presents a problem for later 193PU units where the RFSL C2 hardware is absent as there is then no way to have the basic reversing camera auto select in reverse.

In this firmware we offer an override CAN message for the A6 coding, which can be injected onto the bus by an Arduino, or similar, hardware.

📋 Sent a single byte CAN packet to ID 0x351: 0x52 = Reverse, 0x00 = Forward

📋 An Arduino Sketch for sending the required CAN messages is included in the appendix.

8.1.5 CAN Speed for A6 Mode (Beech - 193PU only)

In addition to the Reverse gear over CAN, the Beech firmware release also adds the ability to send speed pulse counts over CAN to compensate for the missing V-Signal B3 hardware pin on later 193 units.

📋 Sent a three-byte CAN packet to ID 0x351:
Byte 0: 0x52 = Reverse, 0x00 = Forward
Byte 1: Pulse count

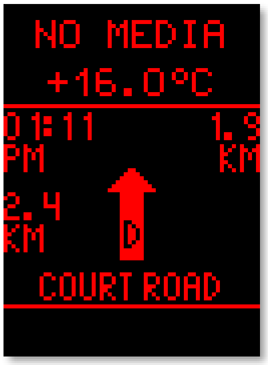
📋 An Arduino Sketch for sending the required CAN messages is included in the appendix.

8.1.6 Lowercase Units on DIS

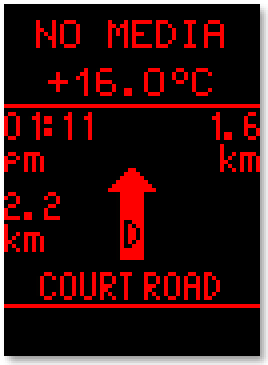
When a distance or time is displayed on the DIS by the navigation system it uses upper case, so either KM/MI or AM/PM.

With this firmware it is possible to switch to displaying these in a more pleasing lower case.

To enable/disable Lowercase Units, see Enabling Features via the Engineering Menus



Normal Units



Lowercase Units

8.1.7 Start Screen Coding

The stock firmware prohibits coding of start screens which do not correspond with the coded vehicle type. However, in some situations it is necessary to code a different vehicle type. For example, an RNS-E installed in a TT 8N needs to be coded to A6 mode.

This firmware introduces the ability to code an override screen in such situations. The required override needs to be coded in addition to the true vehicle variant screen:

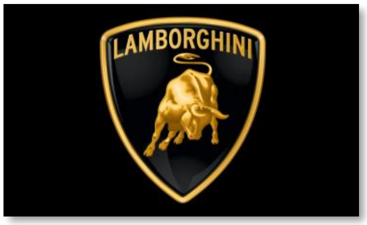
0??x – Override screen	**0xx? – Coded vehicle variant screen**
00 – use vehicle variant screen	0 – default screen
01 – Lamborghini	1 – S3
02 – S3	2 – S4 **
03 – S4 **	3 – RS4 **
04 – RS4 **	4 – TT
05 – TT	5 – R8
06 – R8	6 – Seat *
07 – Seat *	7 – TTS *
08 – TTS *	8 – TTRS *
09 – TTRS *	9 – R8GT *
10 – R8GT *	

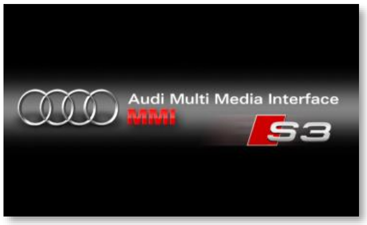
* Not available on 192 units

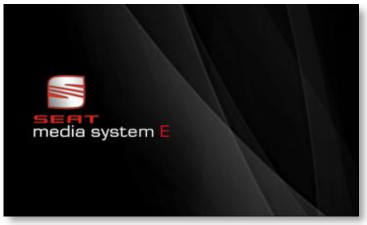
** Not available on 193PU units

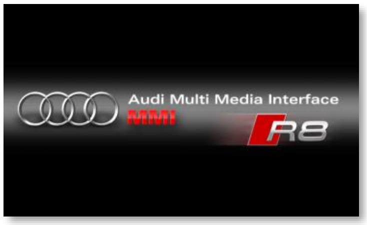
e.g. for a TT 8N code value 050:
05x = override to TT screen
xx0 = from default screen (for A6)

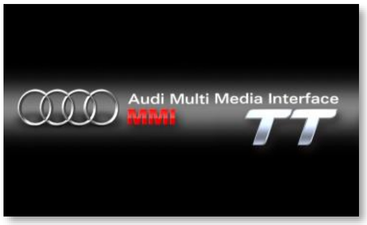
📋 The override screen is coded in the hundreds and tens positions, the true screen for the vehicle variant should be coded normally in the units position
📋 NB: the numbering of the override screens **differs** from the vehicle variant

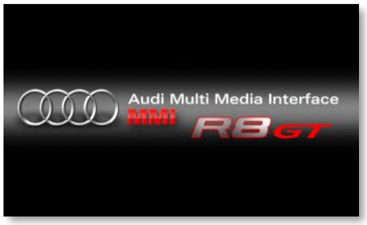










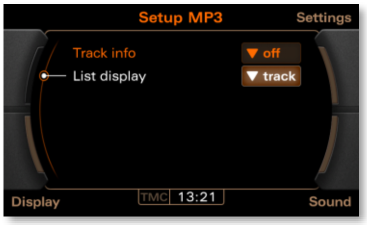


8.1.8 Background Indexing of MP3 ID3 Tags (193PU only)

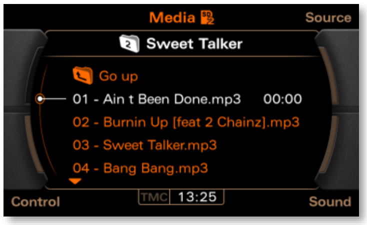
The RNS-E scans the SD Cards for MP3 tracks and extracts the track details from the ID3 tags. You can select either filenames or ID3 tags to be displayed via the MEDIA/CD → SETUP → List Display – file / track option.

The stock RNS-E firmware only performs this indexing task when an MP3 track is NOT currently playing. As soon as a track starts playing, indexing is suspended. This was necessary on the 192 units which had an slower SD Card interface that was not capable of handling the demands of 2 processes (indexer and player) both accessing the card simultaneously.

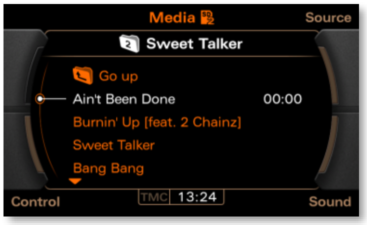
The 193PU units have a greater bandwidth available to the SD Card reader and so this firmware continues to run the indexer in the background. After a short while, with "track" List Display mode enabled, all tracks will be updated on screen with their ID3 tags in place of filenames.



Track info ▼ off
List display ▼ track



Sweet Talker
Go up
01 - Ain t Been Done.mp3 00:00
02 - Burnin Up [feat 2 Chainz].mp3
03 - Sweet Talker.mp3
04 - Bang Bang.mp3



Sweet Talker
Go up
Ain't Been Done 00:00
Burnin' Up [feat. 2 Chainz]
Sweet Talker
Bang Bang

Media Setup Screen "File" view – Filenames "Track" view – ID3 Tags

8.2 Bug Fixes

8.2.1 BCD Clock Fix

On vehicles where the RNS-E was fitted as standard the date/time is set by the cluster over CAN in packed BCD format. e.g. 12:34 is sent as two bytes, 0x12 and 0x34. The RNS-E knows this and displays the time correctly.

Unfortunately, on some other vehicles, e.g. Audi S4 C5/B5 platform, the date/time is sent in raw binary format. This causes the displayed time on the RNS-E to be displayed incorrectly where any hour or minute value is 10 or more.

This firmware contains an automatic fix for this situation. We use the century value to automatically differentiate if the values sent are being transmitted in BCD or raw binary. Values 0x19 and above for the century are assumed to be BCD. Values less than 0x19 (e.g. 19_{10}) are used to imply that the received date/time is in raw binary.

Thanks to Stefan from fis-control.de for recommending this fix.

8.2.2 Red-DIS Protocol Fixes

The RNS-E does not work correctly with certain early vehicles fitted with a Red DIS (for example TT 8N and A4 B7). These are cars fitted with clusters for which the RNS-E was never an official factory fit option.

The symptoms of an incompatible cluster are garbled turn direction graphics on the cluster, and inability to re-select the RNS-E display using the stalk button. When the stalk button malfunctions the DIS display remains blank and it is necessary to stop and restart route guidance to re-enable turn directions.

This firmware includes fixes for both of the above issues. The fix for the stalk button (AKA “Red DIS Bug”) needs to be enabled via the Engineering Mode.

To enable/disable the “Red DIS Bug” fix, see Enabling Features via the Engineering Menus

📋 This new setting takes effect immediately

8.2.3 Fix Abend Processing ID3 Tags

The ID3 indexing task often randomly quits while indexing MP3 tracks. This means that even if MP3 playback is paused SD Cards can often be left only partially indexed.

When this happens tracks in the first few folders are displayed with their ID3 tags, the remainder are left permanently displaying their mp3 filename (or until such time as they are individually played).

This fix ensures that the indexing task remains active. However, note that on 192 units the indexer is still paused while MP3 playback is being performed. See Background Indexing of MP3 ID3 Tags (193PU only)

📋 On 192 units a workaround is either pause MP3 playback, or switch to an alternate audio source (e.g. RADIO), to allow the indexer time to complete reading of all ID3 tags

8.2.4 Fix Bug in Unicode Comment Tags (193PU only)

The 193PU units support reading of ID3v2 tags, however there is a bug in the handling of Unicode COMMent tags. The causes the unit to fallback to using the ID3v1 tags on MP3s with Unicode comment tags present.

This bug has been corrected in this release.

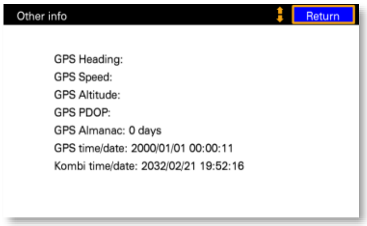
📋 192 units only support ID3v1 tags and so are unaffected by this bug

8.2.5 Fix Display of Kombi Date/Time in Engineering Menus

The display of the current date/time received from the cluster in the Engineering Menu is garbled:
NAV → GPS signal info → more → Kombi time/date

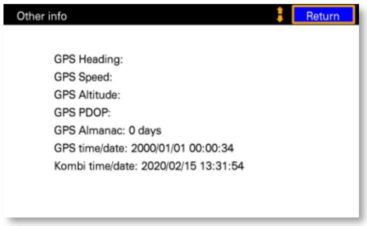
This is because the firmware assumes binary values, but the actual values received from the cluster are in BCD. For example, a value of 0x20 received from the cluster for the 20th day of the month is actually displayed as its decimal equivalent: 32.

This firmware rectifies this coding oversight and correctly decodes BCD date/times.
📋Note this bug was cosmetic – it did not affect data time processing elsewhere



GPS Heading:
GPS Speed:
GPS Altitude:
GPS PDOP:
GPS Almanac: 0 days
GPS time/date: 2000/01/01 00:00:11
Kombi time/date: 2032/02/21 19:52:16

2020 is shown as 2032
13:34 is shown as 19:52



GPS Heading:
GPS Speed:
GPS Altitude:
GPS PDOP:
GPS Almanac: 0 days
GPS time/date: 2000/01/01 00:00:34
Kombi time/date: 2020/02/15 13:31:54

Correct display after update

8.2.6 Reinstated A6 Coding (193PU only)

On the 193PU RNS-E model introduced in late 2009 Audi stopped supporting the A6 mode. This was because by this point production of the A6 C4/4A had ceased. With firmware 0200 Audi also completely removed the ability to code for an A6.

This firmware fixes a number of outstanding bugs that were left in the 193PU firmware for the A6. It also reinstates the ability to code for it.

📋 It is now no longer necessary to downgrade to firmware 0150 to perform an A6 coding

8.2.7 Fix Gala in A6 Mode (193PU only)

At the same time as removing the ability to code for A6 mode, Audi also disabled the correct processing of GALA messages from the CANBUS. The GALA feature no longer has any effect when running a 193PU firmware after SW 0150.

This firmware reintroduces the correct processing of GALA messages when enabled.

📋 To enable GALA see adaptation channel 8 in VCDS.

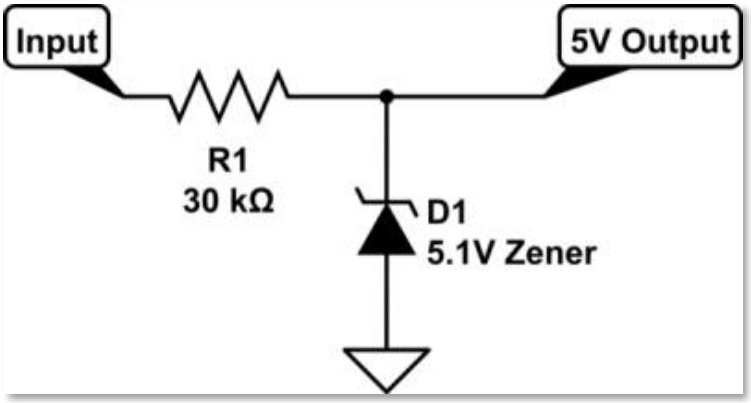
9. Appendix - Arduino CAN Reverse & Speed Injection Sketch

Provided here is a simple Arduino Sketch for injecting suitable CAN messages for out Beech firmware for use in A6 mode 193 units where physical C2 RFSL and B3 V-Signal input pins are unavailable.

```
/*
 * Audi CANBUS RFSL/V-Signal Injector
 *
 * Author: Stuart McConnachie
 * Target: Arduino UNO with MCP2515 CANBUS Shield
 *
 * Injects a special direction and speed CAN packet for RNS-E 193
units operating in A6 mode.
 * This is of use where the RFSL (C2) and V-Signal (B3) pins are not
present (later 193 models).
 * Note that this requires a special firmware to receive the packet:
https://rnse.pcbbc.co.uk
 *
 * Acorn firmware supports a 1 byte CAN packet with the reverse
indicator.
 * Beech firmware also supports a 3 byte packet with reverse AND speed
pulse count.
*/

//CANBUS Library: https://github.com/sandeepmistry/arduino-CAN
#include <CAN.h>

#define DELAY_TIME 100

uint32_t delayPacket;
uint8_t sendCount;

void setup()
{
  // Input from reverse gear
  pinMode(7, OUTPUT);
  digitalWrite(7, 0);

  // Input from reverse gear
  pinMode(6, INPUT);

  // Configure T1 to count V-Signal pulses on pin 5
  pinMode(5, INPUT);
  TCCR1A = 0x00;
  TCCR1B = 0x07;
  TCNT1 = 0;

  // Start debug output
  Serial.begin(9600);
  while (!Serial);
  Serial.println("Audi CANBUS RFSL/V-Signal Injector");

  // Start the CAN bus at 100 kbps
  if (!CAN.begin(100E3))
  {
    Serial.println("Starting CAN failed!");
    while (1);
  }

  // Enable rx
  CAN.filter(0x271,0x7FF);

  // register the receive callback
  CAN.onReceive(onReceive);

  // Start delay between sending CAN packets
  delayPacket = millis();
}


void loop()
{
  // Send a CAN packet every 100ms
  if (millis() - delayPacket >= DELAY_TIME && sendCount > 0)
  {
    // Get V-Signal pulse count
    uint16_t count = TCNT1;

    uint8_t packet[3];

    // Get the reverse gear state
    packet[0] = digitalRead(6) ? 'R' : 0x00;

    // Set the V-Signal pulse count
    packet[1] = count >> 0;
    packet[2] = count >> 8;

    // Send the custom CAN direction and pulse count packet
    CAN.beginPacket(0x351);
    CAN.write(packet, 3);
    CAN.endPacket();

    // Time for next packet
    delayPacket += DELAY_TIME;

    // Send while ignition on
    sendCount--;
  }
}

void onReceive(int packetSize)
{
  // Receive ignition status packet
  if (CAN.packetId() == 0x271)
  {
    uint8_t packet[8];
    uint8_t i = 0;
    while (CAN.available())
    {
      packet[i++] = CAN.read();
    }
    // Ignition on?
    bool ignitionOn = packet[0] >= 0x02;
    digitalWrite(7, ignitionOn);
    if (ignitionOn)
    {
      sendCount = 8;
    }
  }
}
```

We used an Arduino Uno with an MCP2515 EF02037 CAN BUS Shield. The advantage of the Uno is that it can be run directly from the car's 12v supply as it has an onboard voltage regulator:

Note however that the Arduino's inputs are still only 5v, and the reverse gear and speed signals as 12v. You will need to buffer the input signals so as not to damage the Arduino:


Input
5V Output
R1
30 kΩ
D1
5.1V Zener