# Mercedes-Benz model 124 air conditioner (SA code 580) data stream

Lauri Lindholm

July 12, 2023

## Contents

# 1 Disclaimer

The information in this document is provided as is with no warranty of any kind. The information provided is based on reverse engineering efforts which means that its accuracy cannot be guaranteed.

# 2 What?

This document describes the format of the raw "actual value" data stream provided by some Mercedes-Benz model 124 air conditioning control modules.

So far it is to be figured out if only certain modules from certain manufacturers supply this stream, but it is known that not all of them do. It may have been a diagnostic data experiment with the early digital control units or just for manufacturing stage function check.

The terminology used here may differ from the manufacturer's terminology.

# 3 Packet format

A whole packet is made of individual 41 bytes. When capturing data all the way from switch-on, the control module will output 32 bytes of garbage before a consistent data packet is transmitted. The FTDI serial cable used during research may catch one or two bytes of 00 before actual data.

## 3.1 0x00 – temperature setting dial, left INPUT

This is a signed byte in range between -56 (0xc8, 14 °C) and +24 (0x18, 30 °C). The actual temperature value in °C can be calculated with the following formula:

```
actual = (raw + 126) / 5
```

This value indicates the user-requested temperature from the dial position on the control unit's panel. Due to mechanical reasons the temperature on the dial most likely will not match the control unit's value exactly.

## 3.2 0x01 – temperature adjustment target, left INTERNAL

This is a signed byte that indicates the temperature the control module is attempting to adjust to. It is biased by multiple factors, like both user-requested temperatures and the outside air temperature. For actual value, see 0x00 above.

## 3.3 0x02 – temperature setting dial, right INPUT

See 0x00 above.

## 3.4 0x03 – temperature adjustment target, right INTERNAL

See 0x01 above.

### 3.5 0x04 – timer, self-calibration **INTERNAL**

Right after switching on ignition this counter counts from 120 (0x78) down to 0 once in about 10 minutes, ie. in 5 second steps. The counter appears to be used for timing some sort of self-calibration.

1. At 120 (10 min. left) — ignition was switched on, counter starts.
2. At 114 (30 s. after, 9 min. 30 s. left) — First self-calibration.
3. At 96 (2 min. after, 8 min. left) — Automatic air recirculation for intense cooling is enabled.
4. At 60 (5 min. after, 5 min. left) — Second self-calibration.
5. At 0 (10 min. after, timer ends) — Third self-calibration.

### 3.6 0x05 – mixing chamber temperature, left **INPUT**

This unsigned byte ranges between 0 (10 °C) and 243 (0xf3, 70.75 °C). The actual value in °C can be calculated with the following formula:

```
actual = (raw + 40) / 4
```

The mixing chamber temperature affects the feedback loop for the water valve control.

In some terminology this may be inaccurately called heater core temperature.

### 3.7 0x06 – mixing chamber temperature, right **INPUT**

See 0x05.

### 3.8 0x07 – interior air temperature **INPUT**

This signed byte ranges between -128 (0x80, -0.4 °C) and +126 (0x7e, 50.4 °C). The actual value in °C can be obtained with the following formula:

```
actual = (raw + 126) / 5
```

### 3.9 0x08 – exterior air temperature **INPUT**

This signed byte ranges between -64 (0xc0, -32 °C) and 126 (0x7e, 63 °C). The formula for the actual value in °C is as follows:

```
actual = raw / 2
```

### 3.10 0x09 – temperature control, left **INTERNAL**

This signed byte is the difference of dampened interior temperature (see 0x19) and the temperature adjustment target of the respective side. It ranges between -128 (0x80, -25.6 °C) and +127 (0x7f, +25.4 °C). The actual value in °C can be calculated with the following formula:

```
actual = raw / 5
```

Negative values bias towards heating and positive values towards cooling.

### 3.11 0x0a – temperature control, right **INTERNAL**

See 0x09 above.

### 3.12 0x0b – control bias, exterior air temperature **INTERNAL**

This signed byte reacts to change of exterior temperature. The math to its absolute value is not clear, yet. The range of this value is yet to be acquired from the logged data.

As with most other temperature control bias values, negative values bias towards heating and positive values towards cooling.

### 3.13 0x0c – heating control bias, left, fast **INTERNAL**

This unsigned byte ranges between 0 and 255 (0xff). It is used to bias the heating control. 0 calls for no heating and 255 for maximum heating.

This is the fast-reacting value and seems to directly affect water valve control.

### 3.14 0x0d – heating control bias, right, fast **INTERNAL**

See 0x0c.

### 3.15 0x0e – heating control bias, left, slow **INTERNAL**

See 0x0c.

This value follows the value of 0x10 with a dampening. The slowest value seems to affect heating valve control via the temperature feedback loop.

## 3.16 0x0f – heating control bias, right, slow INTERNAL

See 0x0c and 0x0e.

This value follows the value of 0x11 with a dampening.

## 3.17 0x10 – heating control bias, left, mid INTERNAL

See 0x0c.

This value follows the value of 0x0c with a dampening. It's not yet clear how this value affect the heating control.

## 3.18 0x11 – heating control bias, right, mid INTERNAL

See 0x0c.

This value follows the value of 0x0d with a dampening.

## 3.19 0x12 – water valve feedback bias, left INTERNAL

This signed byte ranges between -128 (0x80) and +127 (0x7f). Negative values bias towards opening the valve (heating) and positive values bias towards closing the valve (cooling).

It seems to be biased primarily by the slow control bias (0x0e) and the mixing chamber temprature (0x05).

## 3.20 0x13 – water valve feedback bias, right INTERNAL

See 0x12. For this one the biasing values are from 0x0f and 0x06.

## 3.21 0x14 – water valve solenoid duty cycle, left OUTPUT

This unsigned byte ranges between 0 (0%, valve closed) and 255 (0xff, 100%, valve open).

## 3.22 0x15 – water valve solenoid duty cycle, right OUTPUT

See 0x14.

## 3.23 0x16 – engine coolant temperature INPUT

This signed byte ranges between 5 and 127 (0x7f) within its functional range. If sensor circuit is shorted, the value will be fixed to -126 (0x82). Within the functional range the raw value is the actual value in °C as is.

Engine coolant temperature is used for prevention of overheating of the engine, and possibly for some early heater control until the engine heats up.

## 3.24 0x17 – evaporator temperature INPUT

This unsigned byte ranges between 0 (0 °C) and 126 (0x7e, 63 °C). This temperature value controls the air conditioner compressor request line. The actual value in °C is calculated with the following formula:

```
actual = raw / 2
```

The A/C compressor request turns on when this value is 14 (7 °C) or greater, and off when it falls to 10 (5 °C) or below. The compressor request line is routed through the refrigerant pressure switch to the compressor safety cut-out module.

## 3.25 0x18 – engine overheat protection status INTERNAL

This (most likely) unsigned byte is 0 in normal operation. If the engine coolant temperature sensor circuit is shorted, the value will be 190 (0xbe, -66 if signed). In overheat protection operation the value will count from 64 (0x40) to 103 (0x67).

Engine overheat protection is activated when the engine coolant temperature reaches 122 or above. Once activated, the protection switches off when engine coolant temperature goes down to 117 or below.

During overheat protection the AC compressor operation is inhibited.

## 3.26 0x19 – interior temperature, dampened INTERNAL

See 0x07. This value follows the interior temperature sensor in a dampened manner. It is used for temperature control to avoid unnecessary sudden temperature changes.

## 3.27 0x1a – user input and intense cooling control BITMASK

### 3.27.1 0x1a bit 7 – unused

Appears to be static 0.

### 3.27.2 0x1a bit 6 – intense cooling mode INTERNAL

This bit is set when the control unit operates in intense cooling mode.

### 3.27.3 0x1a bit 5 – user intervention, temperature adjustment, right INTERNAL

This bit is set when the user is making a temperature adjustment. If the adjustment is larger than three units (0.6 °C), the control unit calculates a damping value for the requested temperature.

### 3.27.4 0x1a bit 4 - user intervention, mode change INTERNAL

This bit is set when the user is making a mode change. Since the bit is typically set for a time shorter than the transmission time of a whole data packet, it is most often never seen to change state.

### 3.27.5 0x1a bit 3 – user intervention, temperature adjustment, left INTERNAL

See 0x1a bit 5.

### 3.27.6 0x1a bit 2 – button status: reheat INPUT

This bit indicates the status of reheat mode. When this bit is set, the red LED on the button is lit.

When this mode is enabled, the air conditioning compressor is requested whether cooling is needed or not. The primary use for this is to dry the interior air in case the moisture in the air tends to concentrate on the windscreen or other windows.

### 3.27.7 0x1a bit 1 – button status: economy mode (EC) INPUT

This bit indicates the status of economy mode. When this bit is set, the red LED on the button is lit.

When this mode is enabled, the air conditioning compressor request is inhibited and middle vents are set to bypass heating. Air recirculation is limited to five minutes at a time.

### 3.27.8 0x1a bit 0 – button status: recirculation INPUT

This bit indicates the status of manually requested interior air recirculation. When this bit is set, the red LED on the button is lit.

The requested recirculation is always 100% and is limited to 20 minutes with A/C enabled or 5 minutes in economy mode.

## 3.28 0x1b – circulation timer INTERNAL

This (expected to be) unsigned value contains the amount of minutes until air recirculation is automatically switched off to fresh air.

The countdown starts from 20 (0x14) when air conditioning compressor is enabled and 5 when air conditioning is inhibited.

## 3.29 0x1c – actuator control BITMASK

### 3.29.1 0x1c bit 7 – water circulation pump OUTPUT

This bit is set when the water circulation pump is running.

### 3.29.2 0x1c bit 6 – unused

Appears to be static 0.

### 3.29.3 0x1c bit 5 – unused

Appears to be static 0.

### 3.29.4 0x1c bit 4 – A/C compressor request OUTPUT

This bit is set when the A/C compressor request line is driven. The heater blower must be on for activation and economy mode (EC) must be off.

### 3.29.5 0x1c bit 3 – air recirculation, 80% OUTPUT

This bit is set when the vacuum valve for 80% air recirculation is driven.

### 3.29.6 0x1c bit 2 – air recirculation, 100% OUTPUT

This bit is set when the vacuum valve for 100% air recirculation is driven. Bit 3 is always set together with this one.

### 3.29.7 0x1c bit 1 – radiator blower, stage II OUTPUT

This bit is set when the relay for radiator blower stage II is driven. Radiator blower is switched on at engine coolant temperature sensor value 107 and off at 100.

#### 3.29.8 0x1c bit 0 – temp-control for middle dash vents OUTPUT

This bit is set when the vacuum valve for middle dash vents temperature control flaps is driven.

```
0 = temperature control bypassed
1 = middle vents temperature-controlled
```

When the middle vents are temperature-controlled, they can also be closed to "leak air" state by another vacuum actuator. However, there doesn't seem to be a bit indicating its status.

### 3.30 0x1d – temperature control BITMASK

#### 3.30.1 0x1d bit 7 – recirculation enabled for intense cooling INTERNAL

This is typically set two minutes after switching on ignition.

#### 3.30.2 0x1d bit 6 – self-calibration INTERNAL

When set, the control unit is performing a self calibration. Water recirculation pump is switched off during this time.

#### 3.30.3 0x1d bit 5 – temperature control mode INTERNAL

```
0 = heating
1 = cooling
```

#### 3.30.4 0x1d bit 4 – unused

Appears to be static 1.

#### 3.30.5 0x1d bit 3 - defrost, right INPUT

This bit is set when the temperature control dial is turned all the way to its hot end stop.

#### 3.30.6 0x1d bit 2 - max cooling, right INPUT

This bit is set when the temperature control dial is turned all the way to its cold end stop.

#### 3.30.7 0x1d bit 1 - defrost, left INPUT

See 0x1d bit 3

#### 3.30.8 0x1d bit 0 - max cooling, right INPUT

See 0x1d bit 2

### 3.31 0x1e – temperature dial value, dampened, left INTERNAL

This value follows the value of temperature setting dial. The speed to reach the value is defined by 0x1f.

For range, see 0x00.

### 3.32 0x1f – control value, temp request value change, left INTERNAL

When active, ranges between 4 and 75 (0x4b). Otherwise 0.

### 3.33 0x20 – temperature dial value, dampened, rightINTERNAL

See 0x1e.

### 3.34 0x21 – control value, temp request value change, right INTERNAL

See 0x1f.

### 3.35 0x22 – static 0x00 SYNC:INTERNAL

This and the following six bytes have been used for data stream synchronisation. The actual meaning of these bytes is mostly unknown but they appear to be static data and therefore useful for easy sync.

### 3.36 0x23 – static 0x03 SYNC:INTERNAL

### 3.37 0x24 – static 0x04 SYNC:INTERNAL

### 3.38 0x25 – static 0x01 SYNC:INTERNAL

### 3.39 0x26 – static 0x23 SYNC:INTERNAL

Most likely a version number, possibly hardware revision identifier. The number is 35 in base 10.

### 3.40 0x27 – static 0x02 SYNC:INTERNAL

### 3.41 0x28 – static 0x3b or 0x3c SYNC:INTERNAL

Most likely a version number, possibly software. 59 (0x3b) has been seen on two cases and 60 (0x3c) on a newer car.

## 4 Serial data electricals

The serial data supplied from socket 7 of the diagnostics connector block is basically 8-N-1 at 4,800 bps with about 30 ms gaps between frames. The only major difference to RS-232 or TTL are the signal levels used. See the table below:

| | RS-232 | TTL | MB |
|---|---|---|---|
| mark | -15..-3 V | +5.0 V | +8.0 V |
| space | +3..+15 V | +0.0 V | +0.8 V |

For research purposes the output from the vehicle was converted to TTL by means of a simple circuit of diodes and resistors to use an FTDI TTL-232R-5V "USB to TTL Serial Cable" for data acquisition. This is what the initial decoder programs written in Python were designed around.

## 5 Tested vehicles

The following vehicles were equipped with basic "Tempmatic" air conditioning, SA code 580.

- 124.092 – 320 TE (the original research platform)
- 124.191 – E 300 DIESEL (facelift; this was the exception that had 0x3c as the last sync byte instead of 0x3b)
- 124.193 – 300 TD TURBODIESEL

### 5.1 Known not to work

The following vehicle was equipped with fully automatic air conditioning, SA code 581.

- 124.131 – E 300 DIESEL (US version, SA code 494; facelift) – no data stream)

# 6 32-byte switch-on garbage analysis

This is section is subject to removal when the garbage data is processed and considered useless.

The data stream starts at packet offset 0x09.

Some of the temperature control data below starts consistently but is very much irrelevant this early after switch-on.

```
09 0a  0b  0c 0d 0e 0f 10 11  12 13 14 15
-----------------------------------------
90 F7  FF  A7 FF FF FF FF 49  02 40 FF 14
90 F7  FD  A7 FF FF FF FF 8E  02 40 FF 95
90 F7  FD  A7 FF FF FF FF 75  02 40 FF 6C
90 F7  FD  A7 FF FF FF FF 40  02 40 FF 8E
90 F5  FF  A7 FF FF FF FF 02  02 40 FF 5D
-----------------------------------------
90 Fx  Fx  A7 FF FF FF FF xx  02 40 FF xx
```

Observetions of data between 0x16 and 0x19:

- Engine coolant temperature (0x16) seems to consistently start with an invalid value 0xd0 (208, -48).
- Evaporator temperature consistently starts with invalid value 0xfb (251, -5).
- Engine overheat protection status (0x18) is consistently at value of normal operation.
- Dampened interior air temperature starts at 0xff (-1, 25 °C).

```
16 17 18 19
-----------
D0 FB 00 FF
```

```
D0 FB 00 FF
D0 FB 00 FF
D0 FB 00 FF
D0 FB 00 FF
-----------
D0 FB 00 FF
```

Observations of data between 0x1a and 0x21:

- Bit mask byte at 0x1a has bit 6 (intense cooling) consistently set.
- Circulation timer (0x1b) initialises as 00.
- Bit mask bytes at 0x1c and 0x1d contain uninitialised garbage.
- Temperature dial damping values at 0x1e through 0x20 are consistent but the value at 0x21 is garbage. These will be initialised properly for the next packet.

```
1a 1b 1c 1d  1e 1f 20 21
------------------------
40 00 A6 41  F3 00 00 6E
40 00 46 CB  F3 00 00 7C
40 00 36 8D  F3 00 00 65
40 00 24 11  F3 00 00 65
40 00 22 84  F3 00 00 77
------------------------
40 00 xx xx  F3 00 00 xx
```

The bytes at 0x22 through 0x29 are the sync bytes which are initialised for the following packets.

```
22 23 24 25 27 28 29
--------------------
70 7A 00 92 87 92 59
7B 84 00 6B 84 84 91
64 47 00 86 95 9E A0
64 55 00 85 93 9C A1
72 DE 00 1E B5 B5 C7
--------------------
xx xx 00 xx xx xx xx
```