

Topic:	Gardner Denver – Binary Protocol
Author:	ALL
Date:	2019-05-24

## Inhalt

<b>1. Command Line Demo .....</b>	<b>2</b>
1.1 Error Codes .....	3
<b>2. Easy to use C/C++ functions .....</b>	<b>4</b>
2.1 VC600_init() .....	4
2.2 VC600_SetTemporarySetPoint() .....	4
2.3 VC600_setDefaultSetpoint() .....	4
2.4 VC600_getPressureSetpoint() .....	5
2.5 VC600_getPressure() .....	5
2.6 VC600_Beep() .....	5
2.7 VC600_SetMode() .....	5
2.8 VC600_getSystemState() .....	5
2.9 VC600_startManual() .....	6
2.10 VC600_GetUserLevel() .....	6
2.11 VC600_getRPM() .....	6
2.12 VC600_getAlarm() .....	6

## 1. Command Line Demo

The included demo application uses the function described in 2. For demonstration purposes it can be executed from the command line this:

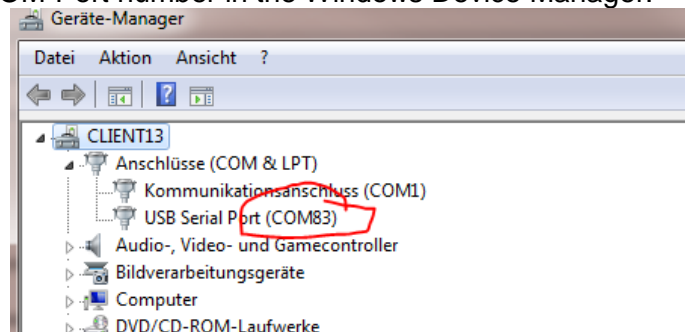
```
VC600_demo.exe COM83 1 200
```

Where the first argument is the COM Port connected to the Device, the second is the command Value (see table below) and the third one the value, if needed.

The command Line Demo also gives Feedback about Errors. An Error code gets shown to the user. See List off Errorcodes to decode them.

#	Parameter/Value	Description
1	Value in mBar	Set temporary set point
2	Value in mBar	Set default set point
5	Time in ms	Beep
10	-	Start manual mode
11	-	Stop
20	-	Get set point
21	-	Get pressure
22	-	Get mode
23	-	Get RPM in %
24	-	Get Alarm

You can see you COM-Port number in the Windows Device Manager:



The program should return success and error messages. Here some examples:

```
X:\VC600-Demo-Dir>VC600_demo.exe COM83 2 500
default set point set to 500
```

```
X:\VC600-Demo-Dir>VC600_demo.exe COM83 10
manual mode started
```

```
X:\VC600-Demo-Dir>VC600_demo.exe COM83 20
set point read: 500mBar
```

```
X:\VC600-Demo-Dir>VC600_demo.exe COM83 1 400
tempoary set point set to 400
```

```
X:\VC600-Demo-Dir>VC600_demo.exe COM83 20
set point read: 400mBar
```

```
X:\VC600-Demo-Dir>VC600_demo.exe COM83 21
preasure read: 510mBar
```

```

X:\VC600-Demo-Dir>VC600_demo.exe COM83 11
stop

X:\VC600-Demo-Dir>VC600_demo.exe COM83 20
set point read: 400mBar

X:\VC600-Demo-Dir>VC600_demo.exe COM83 10
manual mode started

X:\VC600-Demo-Dir>VC600_demo.exe COM83 11
stop

X:\VC600-Demo-Dir>VC600_demo.exe COM82 21
invalid COM-Port
X:\VC600-Demo-Dir>VC600_demo.exe COM83 1
To less arguments!

X:\VC600-Demo-Dir>VC600_demo.exe COM83
To less arguments!

X:\VC600-Demo-Dir>VC600_demo.exe COM83 a
invalid parameter 2 (commando)

X:\VC600-Demo-Dir>VC600_demo.exe COM83 1 a
invalid parameter 3 (value)

X:\VC600-Demo-Dir>Vc600_demo COM88 24
Alarm - Service!

```

## 1.1 Error Codes

Error codes Generated by Software	
Code	Description
2	Header receive failed
3	Data receive failed
4	CRC error
5	Received package was not addressed to PC
6	Received package has invalid Sender-Code
7	Received package has different Command Code than sent
10	Received data has wrong Length
11	Send Error
Error Codes Generated by Firmware	
Code	Description
1001	general parameter error
1003	operation locked -> enter user/admin password to unlock

## 2. Easy to use C/C++ functions

The provided example uses easy to use functions to control VC600. These function are written in C. They should work in a variety of Systems and Devices including PCs and Microcontrollers. Due to C++'s backwards compatibility they can also called from C++ code as shown in the example.

### 2.1 VC600\_init()

This function must be called by program to give references to the needed Communication functions. Communication must be handled by user. See example for one implementation with C++ in Visual Studio 2015.

```
int8_t VC600_init(
    int8_t(*sendBlock)(uint8_t data[], uint16_t length),
    int8_t(*readDataByte)(uint8_t* data),
    int8_t(*flushRX)(void));
```

*sendBlock()	pointer to block send function
*readDataByte()	pointer to read one Byte from RX FIFO function
*flushRX()	pointer to flush RX FIFO function
returns	Error-Code: 0 = no Error

### 2.2 VC600\_SetTemporarySetPoint()

Function for setting a temporary set point. This set point should only be set when VC600 is running in manual Mode. When starting VC600 start always with default set point

This function uses the set point in mBar as integer as parameter. If the second parameter is 0/false the pressure is set absolute. If it is 1/true the set point will get set relative to the actual set point. In this case also a negative pressure can be used. If no Error it returns 0.

```
int16_t VC600_SetTemporarySetPoint (int16_t setpoint, bool absRel);
```

pressure_mBar	pressure set point in mBar
absRel	0-> absolute pressure value 1-> pressure value relative to actual
returns	Error-Code: 0 = no Error

### 2.3 VC600\_setDefaultSetpoint()

This changes the nonvolatile default set point for manual mode. This function is executed slower than VC600\_SetTemporarySetPoint, because a whole set of settings must be read, modified and sent back.

```
int16_t VC600_setDefaultSetpoint(int16_t pressure_mBar);
```

pressure_mBar	pressure set point in mBar
returns	Error-Code: 0 = no Error

## 2.4 VC600\_getPressureSetpoint()

This function gets the actual set point. This function can get called to any time, but if Pump is not active it will return the default set point. The default set point can only be set by configuration tool or by the user on the device itself.

```
int8_t VC600_getPressureSetpoint(uint16_t* pressure_mBar);
```

* pressure_mBar	Pointer for to return set point in mBar
returns	Error-Code: 0 = no Error

## 2.5 VC600\_getPressure()

This function lets you read the actual pressure measured by the internal pressure sensor of VC600. This function can get called to any time.

```
int8_t VC600_getPressure(uint16_t* pressure_mBar);
```

* pressure_mBar	Pointer for to return pressure value in mBar
returns	Error-Code: 0 = no Error

## 2.6 VC600\_Beep()

Function for testing the internal buzzer.

```
int16_t VC600_Beep(int16_t beepMs);
```

beepMs	Duration of sound in ms
returns	Error-Code: 0 = no Error

## 2.7 VC600\_SetMode()

Function to set the operation mode. This function is needed to start manual mode (mode=1) or stop the device (mode=0)

```
int16_t VC600_SetMode(uint8_t mode);
```

mode	Operation Mode 0: stop (no vacuum) 1: manual mode see PressModes_t enum for additional values
returns	Error-Code: 0 = no Error

## 2.8 VC600\_getSystemState()

VC600\_getSystemState() returns the actual mode. This mode can be set by VC600\_SetMode().

```
int16_t VC600_getSystemState(uint16_t *pMode);
```

mode	Operation Mode 0: stop (no vacuum) 1: manual mode see PressModes_t enum for additional values
returns	Error-Code: 0 = no Error

## 2.9 VC600\_startManual()

This function just calls the following functions:

VC600\_SetMode(1)

VC600\_SetTemporarySetPoint (pressure\_mBar, 0)

```
int16_t VC600_startManual(int16_t preassure_mBar);
```

pressure_mBar	pressure set point in mBar
returns	Error-Code: 0 = no Error

## 2.10 VC600\_GetUserLevel()

With this function you can get the actual user level.

```
int16_t VC600_GetUserLevel(int16_t *pLevel);
```

pLevel	User level return: 0:no Acces, 1:user, 2:admin
returns	Error-Code: 0 = no Error

## 2.11 VC600\_getRPM()

Functions return the Motor RPM in percent [0..100]. If the System uses no RPM controlled Motor it returns 0%.

```
int16_t VC600_getRPM(uint16_t *rpm)
```

rpm	Motor RPM in percent
returns	Error-Code: 0 = no Error

## 2.12 VC600\_getAlarm()

Function to read the alarm status. If return is true the maximum Motor operation time is reached and maintenance is needed.

```
int16_t VC600_getAlarm(bool *alarm);
```

alarm	Bool value for alarm status
returns	Error-Code: 0 = no Error