Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

MICRONAS

Application Note

HAL® 8xy, HAL 100x

Programmer Board

APN000046_03EN
Edition Sept. 19, 2013

Contents

Contents, continued

Release Note: **Revision bars indicate significant changes to the previous edition.**

1. Introduction

The Hall programmer board V 5.1 is a general-purpose programming interface which is capable of addressing the programmable Micronas Hall-effect sensor families HAL 8xy and HAL 100x.

The Hall programmer board V 5.1 is fully software compatible to the Hall programmer board V 4.1. Hence, it is possible to replace older boards by V 5.1 while keeping the software programmer.

1.1. Features

– Communication with a PC by serial interface (RS232)

– Data transfer to/from PC board at a Baud rate of up to 57600

– Supervised programming of the Hall-effect sensors

– Output voltage level control

– Up to four Hall-effect sensors can be connected in parallel

– Easy firmware upgrade possible

1.2. Supported Micronas Hall-Effect Sensors

The board supports all programmable Micronas Hall sensors, each having its own special requirements with respect to the programmer board.

Therefore, the board can be run in different operation modes (board modes), which are addressed separately in subsequent sections of this application note.

Table 1–1: Overview Hall sensors and operation modes

Type	Operation Mode	Reference
HAL 805	0 (Emulation V 4.1) 1	Section 4 Section 5
HAL 810	0 (Emulation V 4.1) 1	Section 4 Section 5
HAL 815	0 (Emulation V 4.1) 1	Section 4 Section 5
HAL 817	0 (Emulation V 4.1) 1	Section 4 Section 5
HAL 82x	1	Section 5
HAL 83x	1	Section 5
HAL 855	2	Section 6
HAL 856	3	Section 7
HAL 880	1	Section 5
HAL 1000	0 (Emulation V 4.1) 1	Section 4 Section 5
HAL 1002	1	Section 5

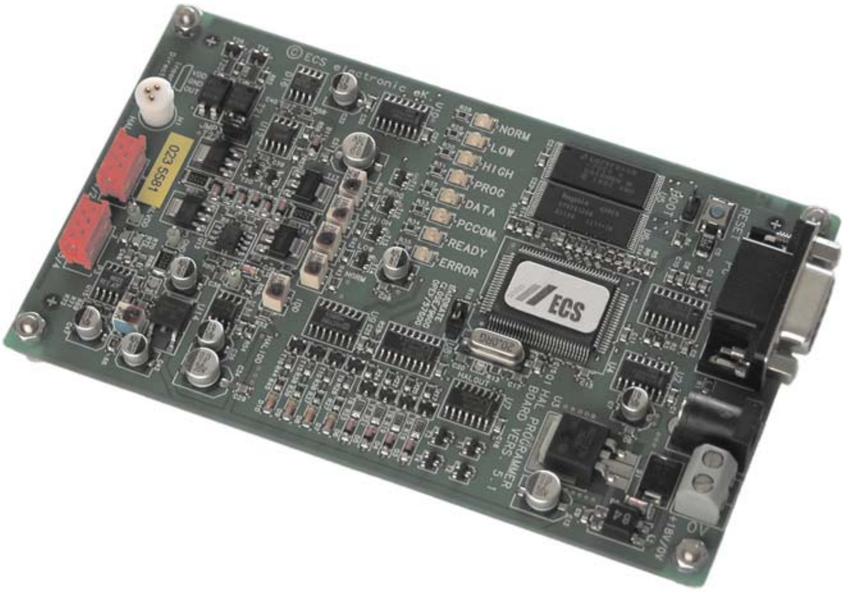

Fig. 1–1: Top view of the Hall programmer board V 5.1

2. Functional Description

The Hall programmer board V 5.1 serves as a communication interface between a PC and the programmable Hall sensor connected to the board.

With a specific programming software, command strings are sent to the board, which generate the serial protocol pattern for the sensor. After sending the protocol, the board reads back the answer of the Hall sensor or – in case of a communication error – generates an error flag.

Depending on the command, the board can send the answer of the Hall sensor back to the PC.

The Hall programmer board V 5.1 can be connected to a serial port of a PC (COM1, COM2, COM3, or COM4) using a 1:1 cable with SUB-D-9 plugs.

2.1. Software

For each of the programmable Hall sensor families, a specific PC software exists. This software provides a graphical user interface based on Microsoft Visual Basic or LabView.

For detailed information on the software, please refer to the software documentation enclosed in the software installation CD.

2.2. Firmware

The firmware of the board is stored in an on-board flash memory. The current version number of the firmware can be read out by sending a “v” command followed by a “t” command (see Table 4–2).

If the Micronas VB software is used, the firmware version number is read out by selecting the menu item:
Help → About.

It is easily possible to do a firmware update (see Section 3.4.1).

2.3. Power-On Self-Test

Firmware version 1.27 or higher provides a power-on self-test procedure:

– After setting the supply voltage or pushing the reset button, the board performs a self-test and measures all voltage levels.

– In case of an error, the status flag is set and the error LED ignites.

– If this happens, disconnect all sensors and try again. If the error persists, please contact your supplier.

2.4. The COM Port Parameters

The basic parameters for the setup of the serial communication are described in Table 2–1.

Table 2–1: COM port parameters

Parameter	Value	Remarks
Baud rate	9600 57600	Jumper 'BaudRate' set Jumper 'BaudRate' open
Data bits	8	
Stop bits	1	
Parity	EVEN	
Flow control	NONE	

2.5. Connectors

Up to four sensors can be connected to the programmer board V 5.1. For this purpose, two 6-pin connectors, HAL 1/2 and HAL 3/4, are supplied.

Note: In case of HAL 810, HAL 856 and HAL 1000 only one sensor can be connected to the board. These sensors do not support the required multi-programming function.

Alternatively, one Hall sensor can be inserted in the three-pin socket HAL beneath the connector HAL 1/2. The pins of this socket are connected parallel to pins 1, 3, and 5 of the connector HAL 1/2.

The male plug (**Amp 215083**) corresponding to the red connectors HAL 1/2 and HAL 3/4 can be ordered from Bürklin with the order no. **58F462**.

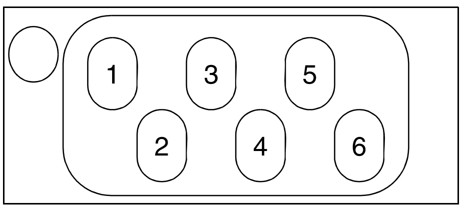

Fig. 2–1: Interface connector HAL 1/2 (HAL 3/4).

2.5.1. Pinning of the Interface Connector

The pinning of the interface connector is described in Table 2–2.

Table 2–2: Pinning HAL Interface

Pin No.	Description
1	Sensor input HAL 1/2: V_{DD} Sensor 1 HAL 3/4: V_{DD} Sensor 3
2	Sensor input HAL 1/2: V_{DD} Sensor 2 HAL 3/4: V_{DD} Sensor 4
3, 4	Common sensor ground
5	Sensor output HAL 1/2: V_{OUT} Sensor 1 HAL 3/4: V_{OUT} Sensor 3
6	Sensor output HAL 1/2: V_{OUT} Sensor 2 HAL 3/4: V_{OUT} Sensor 4

2.6. LED Description

Table 2–3: LED description

LED	Description	Remarks
NORM	Normal operating voltage (V_{DD})	5.0 V calibrated
LOW	Low level of telegram (V_{DD})	5.5 V calibrated
HIGH	High level of telegram (V_{DD})	8.0 V calibrated
PROG	Programming voltage level	12.5 V calibrated
READY	On when board ready for operation	
DATA	High level of telegram (V_{OUT})	
PCCOM	Receive data from PC	
ERROR	Error flag	Status = 1

2.7. Error Flag Codes

Each response sent from the programmer board to the PC contains a status or error flag.

Table 2–4: Error flag description

Error Flag	Description
0	No error; status OK
1	Unspecified system error
2	Output low-level detection failure
3	Missing acknowledge (ACK)
4	ACK time-out failure
5	Bit time < 1 ms
6	VPROG out of range

Note: For safe programming, the evaluation of the error flags is mandatory.

3. Usage of the Hall Programmer V 5.1

3.1. Tips for Safe Programming of the Sensors

The Hall programmer board V 5.1 generates all voltage levels according to the specification of the serial protocol.

V_{PROG} external generation

In case of the sensors HAL805, HAL815, HAL817, HAL810, HAL824, HAL825, HAL830, HAL835, HAL880, HAL1000 and HAL1002 the programming voltage V_{PROG} (which is necessary for the permanent storing of the data in the EEPROM memory) has to be generated externally, i.e. by the programmer board. If V_{PROG} is out of the specification limits, the programming of the EEPROM memory may be insufficient and the reliability of the sensor may be reduced. Hence, the Hall programmer board V 5.1 is equipped with an on-board A/D-converter which reads back the programming voltage V_{PROG}. The A/D-converter is polled every 80 µs.

If V_{PROG} drops below 12.4 V, an error flag is set and the ERROR LED on the programmer board ignites. Please ensure that your software evaluates the error flags (see Section 2.7).

V_{PROG} internal generation

In case of the sensors HAL855 and HAL856, the programming voltage is generated internally. The supply voltage of the Hall sensor during programming must be 5.0 V.

The width of both pulses (erase and prom) is defined by the programmer board. In case a customer-specific software is implemented, please ensure that the programming time is set to 100 ms.

We recommend supervising the programming process and verification of the results after programming the sensors (see Table 3–1).

Note: The programming voltage V_{PROG} is measured at the output of the Hall programmer board. However, this measurement cannot consider any voltage drop which may occur between the Hall programmer board and the Hall sensor.
The voltage drop between the programmer board and the Hall sensor must not exceed 50 mV.
Please ensure that the Hall sensor and the programmer board are connected low resistively to a common ground.

Note: Electrostatic discharges (ESD) may disturb the programming pulses. Please take precautions against ESD.

Table 3–1: Check list

Check Item	When
Status/error flag	After transmission of each write, read, or store command
Programming voltage	After sending a store command (ERASE and PROM)
Programming pulse width	Evaluation of software after implementation
Verify written data	After finishing the programming sequence
Lock function	After lock (Note: The Lock function is active after the next power-on reset.)
Analog test (functional test)	After locking

3.2. Recommended Operating Conditions

Functional operation at conditions other than those listed below is not implied and may result in insufficient programming of the connected sensor and may cause damage to the Hall programmer board.

Symbol	Parameter	Min.	Typ.	Max.	Unit	Conditions
V_{SUP}	Supply Voltage (DC)	17.0	18.0	20.0	V	
V_{ACp-p}	Supply Voltage AC Ripple	–	0	0.150	V_{p-p}	
I_{SUP}	Supply Current	50	120	300	mA	
C_L	Load Capacitance	–	–	2000[1)]	nF	
I_{OUT}	Continuous Output Current	–	–	60	mA	
T_A	Ambient Temperature Range	10	25	50	°C	

1) Example: If four sensors are connected to the programmer board, blocking capacitors of 500 nF between V_{DD} and GND of each sensor can be applied.

3.3. Recommended Wiring

We recommend connecting the application to the board using shielded wires.

In order to minimize the risk of electromagnetic disturbances, the cable should be as short as possible.

Note: Especially in noisy environments beneath power switches, electromagnetic actuators, and the like, EMI-compliant layout of the wiring is mandatory.

For recommended cable parameters, please refer to Table 3–2.

Note: The programmer board reads back the programming voltage every 80 μs in order to be able to detect short spikes on the output voltage line. If cables other than those specified in Table 3–2 are used, such spikes may not be detected by the board.

3.4. Maintenance and Calibration

We recommend sending the programmer board back to the supplier for maintenance and calibration of the voltage levels after one year of operation.

The Hall programmer board must not be maintained or repaired by the customer. In case of any problems or defects, please contact your supplier.

WARNING: Do not modify any part of the Hall programmer board V 5.1, nor readjust any trimming potentiometer. Otherwise, the board may be damaged, the sensor programming may be insufficient, and the reliability of the sensor reduced.

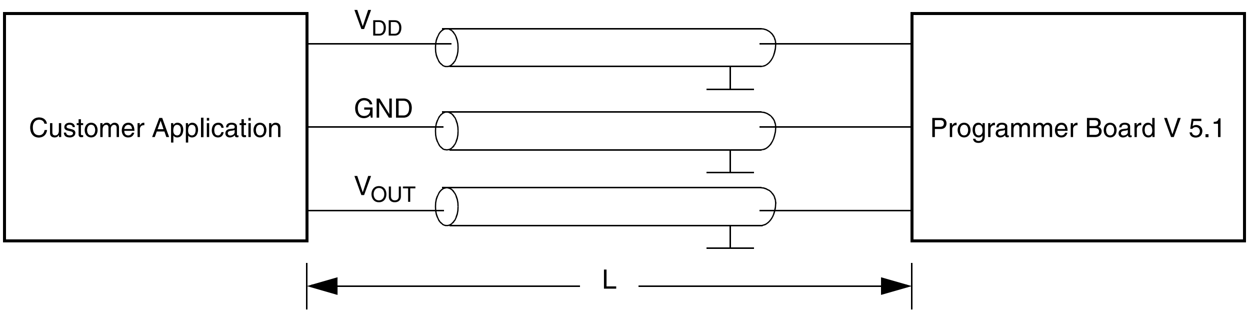

Fig. 3–1: Recommended wiring – schematic sketch

Table 3–2: Recommended cable parameters

Symbol	Parameter	Min.	Typ.	Max.	Unit	Conditions
R_0	Ohmic Resistance per Wire	–	1	5	Ω	$I \leq 10$ mA
C_0	Capacitance	–	80	120	pF	
Z	Impedance	–	50	–	Ω	
L	Length	–	–	1	m	

3.4.1. Firmware Update

The HAL Programmer V5.1 is equipped with a flash-memory. Thus, firmware-upgrades can be done easily by flashing the new firmware into this memory. For this purpose the software-tool-kit ‘FLASH IT’ is provided.

1. Unzip the package ‘Flashit.zip’ into a suitable working directory. When unzipping with ‘WinZip’ a sub-folder ‘\Flashit’ will be generated.
2. Close the jumper ‘BOOT’ beneath the ‘Reset’-button of the HAL Programmer Board.
3. Push the ‘Reset’-button.
4. Open the jumper ‘BOOT’ again.
5. Start the executable ‘Bootload.exe’ by double-clicking the icon in \Flashit.
6. After hitting the  icon, the flash process starts.

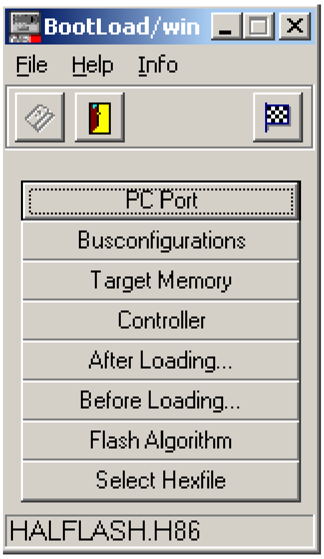

Fig. 3–2: Flash procedure

Note: The flash process lasts for about 10 – 15 min. After the process has finished, the board will be reset automatically and the new firmware will start working.

Note: The firmware to be loaded into the flash-memory is called ‘HALFLASH.H86’. It must be stored in the same folder where the executable is located.

4. Operation Mode 0

Emulation of the Programmer Board V 4.1:
HAL 805, HAL 815, HAL 817, HAL 810, and HAL 1000

This operation mode is the default mode. It is invoked after power-on or pushing the reset button. Alternatively, this operation mode can be set by sending the “j0” command.

Communication: PC → prog. board

Section 4.1 and Section 4.2 contain a detailed description of communication between the PC and the programmer board V 5.1. In particular, all board commands available in the operation mode “j0” are listed.

Communication: prog. board → sensor

The communication between the programmer board and the Hall sensor is described in Section 4.3

Data formats

The data formats for communication between both the PC and the board, and the board and the sensor are summarized in Section 4.4

Programming procedure

Finally, Section 4.5 depicts the programming procedure and gives an example of a complete programming sequence.

4.1. Definition of the Protocol

The protocol and the commands in the emulation mode are exactly the same as those of the programmer board V 4.1.

The general syntax is as follows (blanks are given for better readability only):

PC → Board:
STX BCMD [CMD CP ADR AP] [DAT3 DAT2 DAT1 DAT0 DP] ETX

Board → PC:
STX STATUS DAT3 DAT2 DAT1 DAT0 DP ETX

The characters in brackets [...] are optional, depending on the board command BCOM.

Table 4–1: Description of the characters

Character	Description
STX	ASCII character 2
ETX	ASCII character 3
BCMD	Board command
CMD	Command
CP	Command parity
ADR	Address
AP	Address parity
DAT3 DAT2 DAT1 DAT0	Data characters, each encoding 4 bit in HEX format. Example: 0 1 f 0 (HEX) = 496
DP	Data parity
STATUS	Status/error flag

4.2. The Board Commands BCMD

The board commands activate functions of the firmware. The characters following the BCMD are the parameters of this function. Some functions serve to control the board operation and do not address the Hall device.

The read, write, program, and lock functions communicate with the Hall device. The parameters of these functions are converted into a serial telegram and is sent to the connected sensor.

Table 4–2: Description of the board commands

BCMD	Parameters	Description
n	–	Switch V_{DD} on
o	–	Switch V_{DD} off
z[t]	t = 10...255 (as ASCII character) default t = 162 HAL8x5, HAL810, HAL1000: set t = 85	Set bit time slow mode bit time = t × 0.02 ms default bit time = 3.24 ms HAL8x5, HAL810, HAL1000: set bit time = 1.7 ms
u[t]	t = 10...255 (as ASCII character) HAL8x5, HAL810, HAL1000: set t = 200	Set programming pulse width width = t × 0.50 ms HAL8x5, HAL810, HAL1000: set width = 100 ms
a[t]	t = 10...255 (as ASCII character) recommended t = 25	Select sensor A, select pulse width = t × 4.0 µs recommended pulse width = 100 µs
b[t]	t = 10...255 (as ASCII character) recommended t = 25	Select sensor B, select pulse width = t × 4.0 µs recommended pulse width = 100 µs
e[p]	p = CMD CP ADR AP DAT3 DAT2 DAT1 DAT0 DP	Write HAL and echo data to PC
q[p]	p = CMD CP ADR AP	Read HAL and echo data to PC
m[p]	p = CMD CP ADR AP	Program HAL and echo program voltage to PC
l[p]	p = CMD1 CP1 ADR1 AP1 CMD2 CP2 ADR2 AP2	Lock HAL and echo program voltage to PC
v	–	Request firmware version
t	–	Request status and echo data to PC
j[p]	p = 0...3 (as HEX)	Switch board operation mode 0 = EMU Board V 4.1 (default) 1 = HAL805, HAL815, HAL817, HAL810, HAL82x, HAL83x, HAL880, and HAL100x 2 = HAL855 3 = HAL856

Examples:

(Blanks are given for better readability only)

– power on:
  STX n ETX
– set bit time slow CHR$(85) = 'U':
  STX z U ETX
– read register no. 2:
  STX q 2 0 2 1 ETX
– write register no. 2:
  STX e 3 1 2 1 0 0 0 A 1 ETX
– store sequence:
  STX u CHR$(200) ETX (set programming time)
  STX m 5 1 1 1 ETX (ERASE)
  STX m 4 0 1 1 ETX (PROM)
– deactivate sensors:
  STX e 3 1 F 0 0 8 0 F 0 ETX
– select sensor A:
  STX a CHR$(25) ETX
– lock sensors
  STX u CHR$(200) ETX (set programming time)
  STX l 7 0 6 0 5 1 1 1 ETX (LOCK-ERASE)

4.2.1. Monitoring of the Programming Voltage

In case of ERASE, PROM, and LOCK, the programming pulse voltage is measured by the board, and a data string is sent back to the PC. The relation between the data and the programming voltage is

$$V_{PROG} = \frac{DAT}{4095} \cdot 6 \cdot 2.485\ \text{V}$$

If the programming pulse voltage is out of the specification limits, the status bit is set to “1”.

Examples:

– V_{PROG} = 12.15 V, Data board → PC:
  STX 1 0 D 0 A 0 ETX
  (DAT = 0 D 0 A)
– V_{PROG} = 12.50 V, Data board → PC:
  STX 0 0 D 6 9 0 ETX
  (DAT = 0 D 6 9)

4.3. Programming of the Sensor

4.3.1. Definition of Programming Pulses

The sensor is addressed by modulating a serial telegram on the supply voltage. The sensor answers with a serial telegram on the output pin.

The bits in the serial telegram have a different bit time for the V_{DD}-line and the output. The bit time for the V_{DD}-line is defined through the length of the Sync bit at the beginning of each telegram. The bit time for the output is defined through the Acknowledge bit.

A logical “0” is coded as no voltage change within the bit time. A logical “1” is coded as a voltage change between 50% and 80% of the bit time. After each bit, a voltage change occurs.

4.3.2. Definition of the Telegram

Each telegram starts with the Sync bit (logical 0), 3 bits for the Command (COM), the Command Parity bit (CP), 4 bits for the Address (ADR), and the Address Parity bit (AP).

There are 4 kinds of telegrams:

Write a register

After the AP bit, follow 14 Data bits (DAT) and the Data Parity bit (DP). If the telegram is valid and the command has been processed, the sensor answers with an Acknowledge bit (logical 0) on the output (see Fig. 4–2).

Read a register

After evaluating this command, the sensor answers with the Acknowledge bit, 14 Data bits, and the Data Parity bit on the output (see Fig. 4–3).

Programming the EEPROM cells

After evaluating this command, the sensor answers with the Acknowledge bit. After the delay time t_w, the supply voltage rises up to the programming voltage (see Fig. 4–4).

Activate a sensor

If more than one sensor is connected to the supply line, selection can be done by first deactivating all sensors. The output of all sensors will be pulled to ground by the internal 10 kΩ resistors. With an activate pulse on the appropriate output pin, an individual sensor can be selected. All following commands will only be accepted from the activated sensor (see Fig. 4–5).

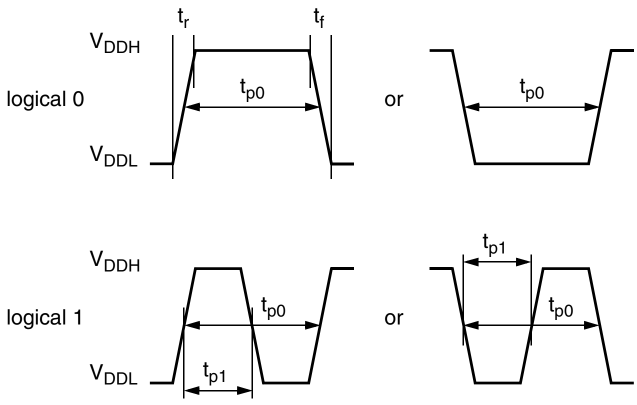

Fig. 4–1: Definition of logical 0 and 1 bit

Table 4–3: Telegram parameters

Symbol	Parameter	Pin	Min.	Typ.	Max.	Unit	Remarks
V_{DDL}	Supply voltage for low level during programming	1	5	5.6	6	V	
V_{DDH}	Supply voltage for high level during programming	1	6.8	8.0	8.5	V	
t_r	Rise time	1			0.05	ms	
t_f	Fall time	1			0.05	ms	
t_{p0}	Bit time on V_{DD}	1	1.7	1.75	1.8	ms	t_{p0} is defined through the Sync bit
t_{pOUT}	Bit time on output pin	3	2	3	4	ms	t_{pOUT} is defined through the Acknowledge bit
t_{p1}	Voltage change for logical 1	1, 3	50	65	80	%	% of t_{p0} or t_{pOUT}
V_{DDPROG}	Supply voltage for programming the EEPROM	1	12.4	12.5	12.6	V	
t_{PROG}	Programming time for EEPROM	1	95	100	105	ms	
t_{rp}	Rise time of programming voltage	1	0.2	0.5	1	ms	
t_{fp}	Fall time of programming voltage	1	0		1	ms	
t_w	Delay time of programming voltage after Acknowledge	1	0.5	0.7	1	ms	
V_{act}	Voltage for an Activate pulse	3	3	4	5	V	
t_{act}	Duration of an Activate pulse	3	0.05	0.1	0.2	ms	

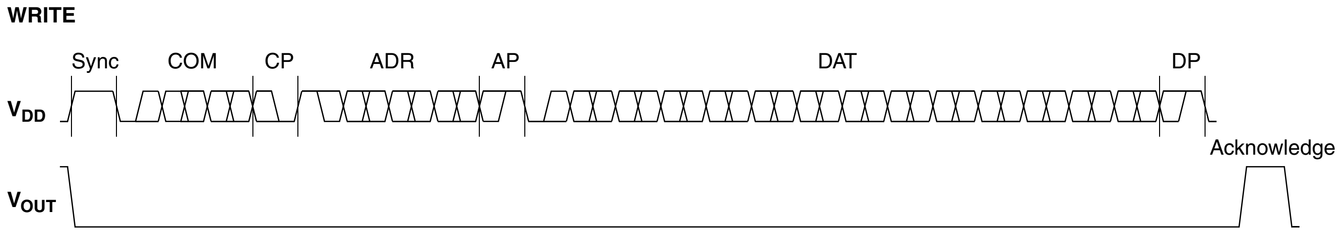

Fig. 4–2: Telegram for coding a Write command

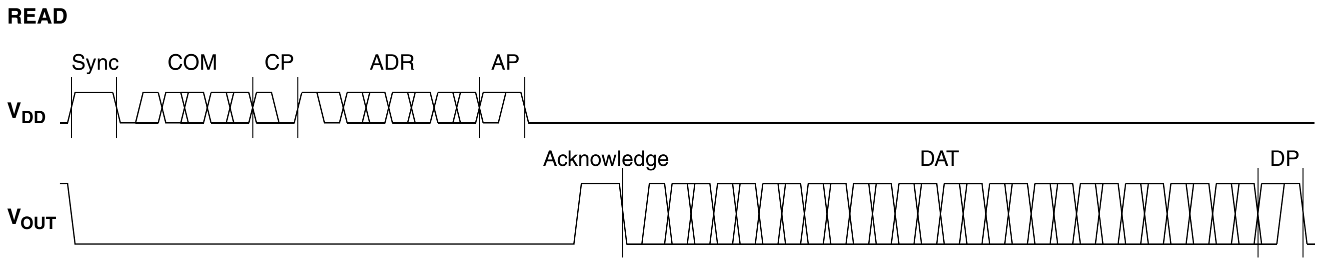

Fig. 4–3: Telegram for coding a Read command

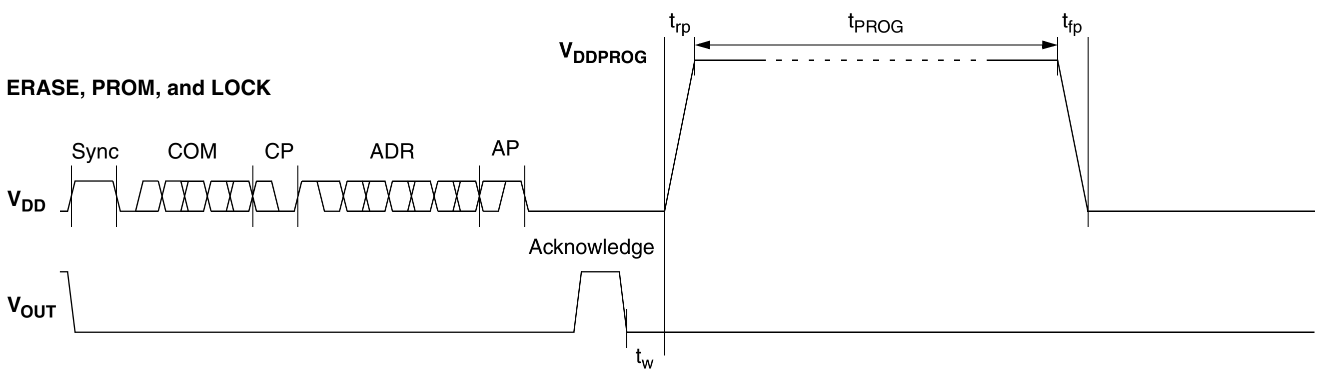

Fig. 4–4: Telegram for coding the EEPROM programming

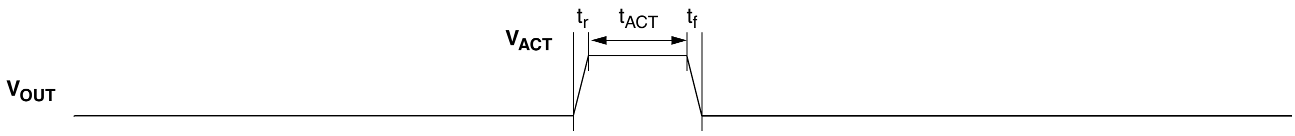

Fig. 4–5: Activate pulse

4.3.3. Telegram Codes

Sync Bit

Each telegram starts with the Sync bit. This logical “0” pulse defines the exact timing for t_{p0}.

Command Bits (COM)

The Command code contains 3 bits and is a binary number. Table 4–4 shows the available commands and the corresponding codes for the HAL 805, HAL 815, HAL 817 and HAL 1000.

Command Parity Bit (CP)

This Parity bit is “1” if the number of zeros within the 3 Command bits is uneven. The Parity bit is “0”, if the number of zeros is even.

Address Bits (ADR)

The Address code contains 4 bits and is a binary number. Table 4–5 shows the available addresses for the HAL 805, HAL 815, HAL 817 and HAL 1000 registers.

Address Parity Bit (AP)

This Parity bit is “1” if the number of zeros within the 4 Address bits is uneven. The Parity bit is “0” if the number of zeros is even.

Data Bits (DAT)

The 14 Data bits contain the register information.

The registers use a different number formats for the Data bits. These formats are explained in Section 4.3.4

In the Write command, the last bits are valid. If, for example, the TC register (6 bits) is written, only the last 6 bits are valid.

In the Read command, the first bits are valid. If, for example, the TC register (6 bits) is read, only the first 6 bits are valid.

Data Parity Bit (DP)

This parity bit is “1” if the number of zeros within the binary number is even. The parity bit is “0” if the number of zeros is uneven.

Acknowledge

After each telegram, the output answers with the Acknowledge signal. This logical “0” pulse defines the exact timing for t_{pOUT}.

Table 4–4: Available commands

Command	Code	Explanation
READ	2	read a register
WRITE	3	write a register
PROM	4	program all nonvolatile registers (except the lock bits)
ERASE	5	erase all nonvolatile registers (except the lock bits)
LOCK	7	lock the whole device and switch permanently to the analog-mode

4.3.4. Number Formats

Binary number: The most significant bit is given as first, the least significant bit as last digit.

Example:

`101001` represents 41 decimal.

Signed binary number: The first digit represents the sign of the following binary number (1 for negative, 0 for positive sign).

Example:

`0101001` represents +41 decimal
`1101001` represents –41 decimal

Two's-complementary number: The first digit of positive numbers is "0", the rest of the number is a binary number. Negative numbers start with "1". In order to calculate the absolute value of the number, calculate the complement of the remaining digits and add "1".

Example:

`0101001` represents +41 decimal
`1010111` represents –41 decimal

Table 4–5: Available register addresses

Register	Code	Data Bits	Format	Customer	Remark
CLAMP LOW	1	10	binary	read/write/program	low clamping voltage
CLAMP HIGH	2	11	binary	read/write/program	high clamping voltage
VOQ	3	11	two compl. binary	read/write/program	
SENSITIVITY	4	14	signed binary	read/write/program	
MODE	5	6	binary	read/write/program	range and filter settings
LOCK	6	1	binary	lock	lock bit
ADC-READOUT	7	14	two compl. binary	read	
TC	11	6	signed binary	read/write/program	
TCSQ	12	5	binary	read/write/program	
DEACTIVATE	15	12	binary	write	deactivate the sensor