USB ADC

Features

- Compact USB ADC board for measurement and control applications
- Eight bipolar voltage inputs -2,047 V … + 4,095 V (single-ended)
- Paired channels useable for differential measurements (+/-4,095V)
- Resolution better than 12 bits (1mV per step)
- Max. 400 samples/sec.(2ms per channel)
- Voltage reference internal (4,095V) or external
- Dimensions approx.. 70 x 45 x 15 mm
- Power supply (+5V) either USB or external
- Convenient screw-terminal connector
- USB chip set: CH341A
- Integrated I²C/TWI master interface
- System requirements XP, 2000, Vista, WIN 7, 32 Bit
- Suitable for ProfiLab 4.0.software
- Easy programming Interface with ActiveX control (OCX)
- Programming examples in C++, Delphi und Visual Basic.

Download link for manual, driver and examples

http://www.abacom-online.de/div/setup_usb_ADC.exe

Installation

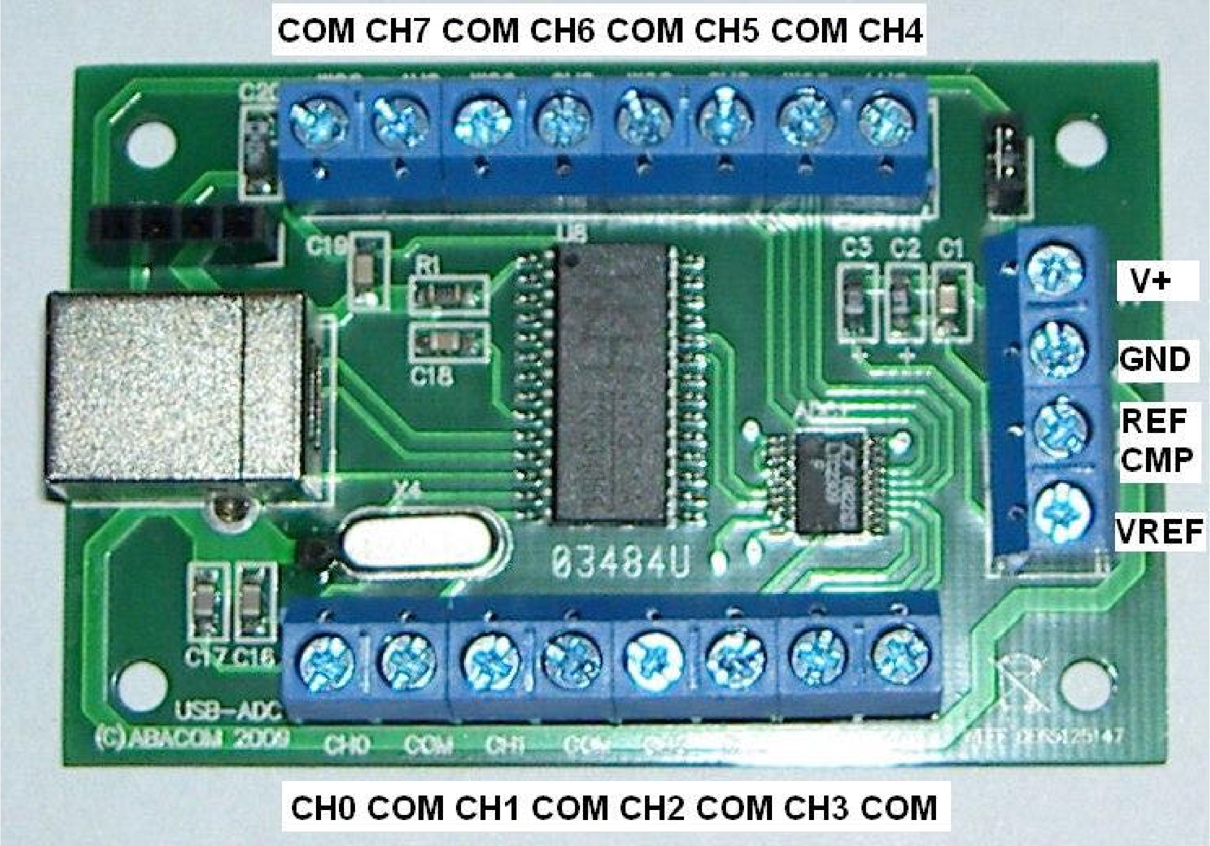
- Before connecting the device to the USB download and install setup software please.
- After that connect the device to the USB and follow the Windows Plug & Play instructions.
- The device is now ready for use. Further information on how to use the board can be found in the manual installed with the software.

Screw terminals

The module is equipped with screw terminal contacts, which make is easy to connect peripheral circuits.

- Voltage inputs CH0 … CH7
- Common potential COM for voltage inputs
- GND: Ground (connected to PC ground)
- VREF: LTC2309 internal Vref (+2,5V)
- REFCMP: Reference voltage for measurement
- V+: USB power supply voltage (+5V)

The terminals CH0 to CH7 represent the ADC voltage inputs.


COM CH7 COM CH6 COM CH5 COM CH4
V+
GND
REF CMP
VREF
CH0 COM CH1 COM CH2 COM CH3 COM
USB-ADC
(C) ABACOM 2009
03484U

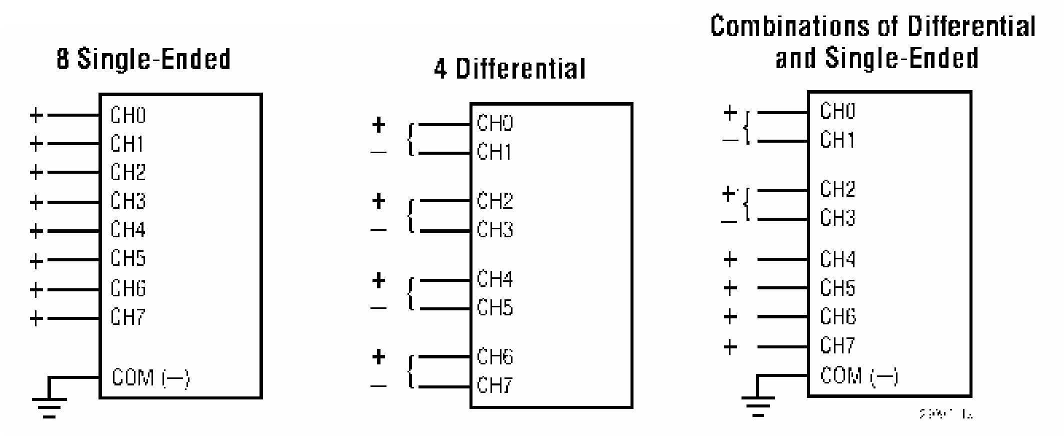
Connecting signals to the ADC make sure signals are within the „Absolute Maximum Ratings“ of the LTC2309 to prevent damage.

The voltage at CH0…CH7, COM and VREF must not exceed the following limit at any time: (GND) – 0,3V to (V+) + 0,3V

This means that input signals must never be negative (related to PC ground (GND)) and must never exceed the USB supply voltage (V+).

Single-ended measurements

Voltages are measured relatively to a common potential (COM). To make wiring easier a COM connection is available beside each voltage input. All COM connections are bound together internally. The **single ended** measurement results are requested through the **software channels CH0 to CH7**.



Differential measurements

Pairs of inputs can be used for differential measurements. The pairs are (CH0-CH1), (CH2-CH3), (CH4-CH5) and (CH6-CH7), where the lower input number of a pair is the non-inverting input and the higher number is the inverting input. The results of **differential measurements** are requested through the **DF01, DF23,DF45 and DF67**.

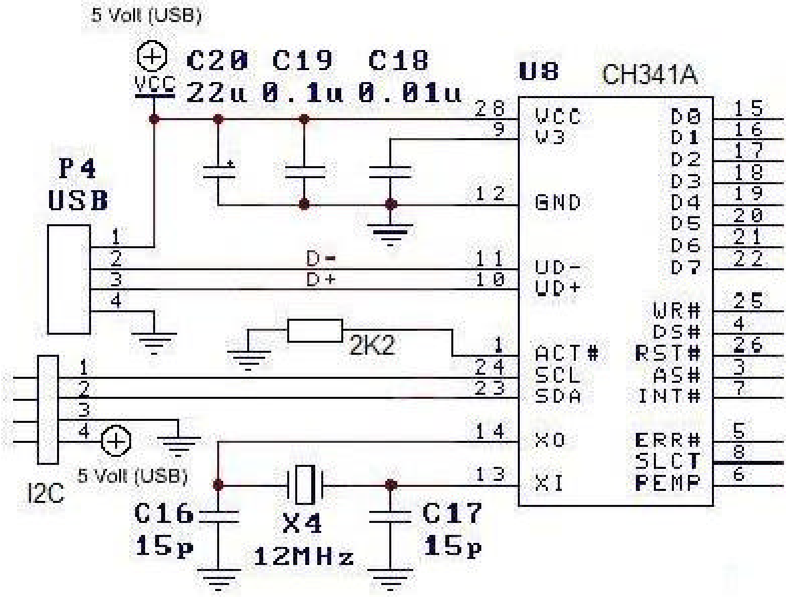
All software channels (single-ended and differential) can be request any time and independently from the actual input circuitry that is being used.

Software channel name	Measurement	Signal injection +	Signal injection -	Range*
ADC_CH0	Single-ended	CH0	COM	-2,047 V…+4,095 V
ADC_CH1	Single-ended	CH1	COM	-2,047 V…+4,095 V
ADC_CH2	Single-ended	CH2	COM	-2,047 V…+4,095 V
ADC_CH3	Single-ended	CH3	COM	-2,047 V…+4,095 V
ADC_CH4	Single-ended	CH4	COM	-2,047 V…+4,095 V
ADC_CH5	Single-ended	CH5	COM	-2,047 V…+4,095 V
ADC_CH6	Single-ended	CH6	COM	-2,047 V…+4,095 V
ADC_CH7	Single-ended	CH7	COM	-2,047 V…+4,095 V
ADC_DF01	Differential	CH0	CH1	-4,095 V…+4,095 V
ADC_DF23	Differential	CH2	CH3	-4,095 V…+4,095 V
ADC_DF45	Differential	CH4	CH5	-4,095 V…+4,095 V
ADC_DF67	Differential	CH6	CH7	-4,095 V…+4,095 V

The connections VREF and REFCMP are explained in chapter „ADC and reference voltage”. Additional information can be taken from the LTC2309 datasheet.

* Due to software optimization (automatic range selection) a resolution of **1mV/step** is reached **for all channels**, which is even better, than the resolution that can be calculated from the LTC2309 ranges and its 12 data bits.

USB interface

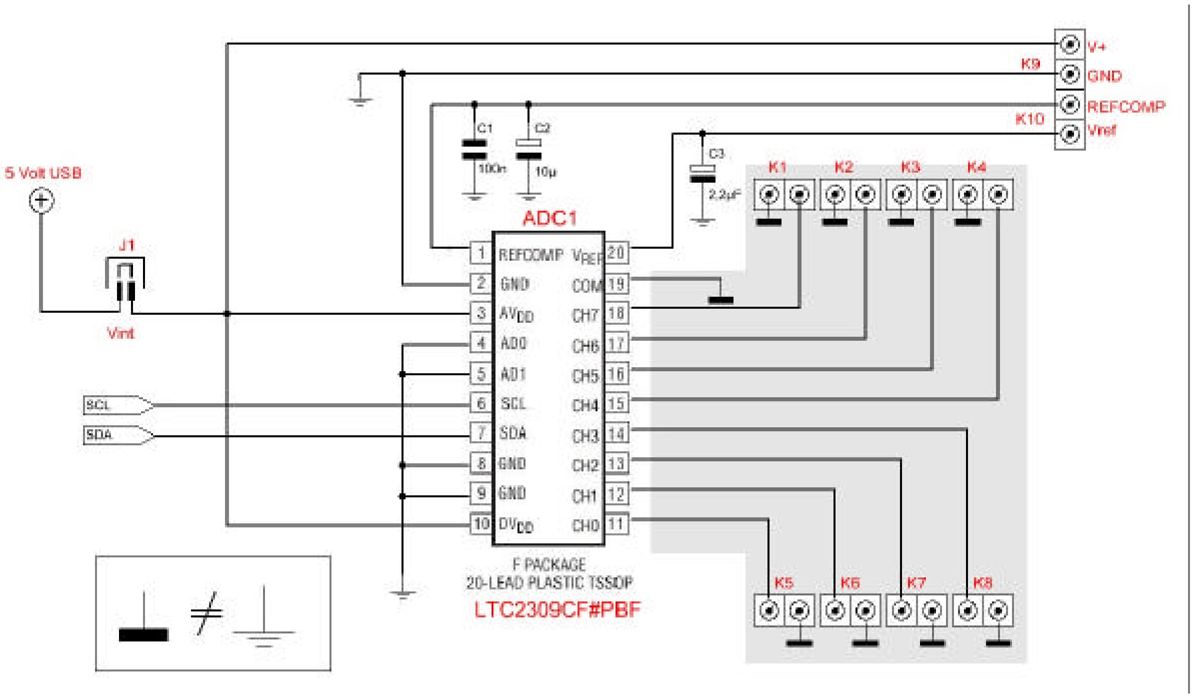


Die Ansteuerung des ADC erfolgt intern über eine I²C-Datenverbindung. Dabei belegt der LTC2309 die I²C-Chipadresse 08hex.

Der I²C-Datenbus steht über eine 4-polige Stiftbuchse für Schaltungserweiterungen zur Verfügung, so dass das ADC-Modul zusätzlich als PC-nach-I²C-Adapter genutzt werden kann.

The ADC is internally controlled through a I²C bus (TWI). A CH341A chip is used as USB-to-I²C bridge.The I²C chip address 08hex is reserved for the LTC2309. Additional I²C chips can be connected if necessary.

ADC and reference voltage



The analogue part consists of a high-quality ADC (Linear Technologies LTC2309) with internal voltage reference.

Internal reference voltage

The screw terminal VREF is connected to the internal voltage reference (2,5V; 8KOhm source resistance). The LTC2309 internally gains this voltage to 4,095V connected to terminal REFCMP. The voltage at REFCMP (4,095V) is the reference for measurement, which means that the ADC reaches full scale as an input voltage rises to the voltage at REFCMP.

External reference voltage

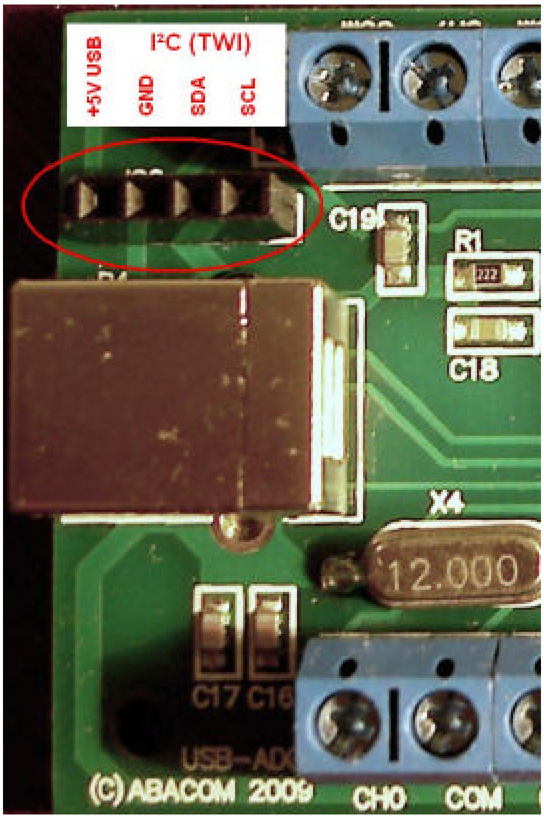
Instead of the internal reference an external voltage can be used. For that purpose VREF must be connected to GND. Now a external voltage can be supplied to the REFCMP terminal.

Important: 0 < REFCOMP < V+

Power supply
While jumper J1 is plugged the V+ terminal is connected to the USB supply voltage (+5V), which takes over the internal power supply of the ADC as well. Supplying external circuits from this pin remember that USB current is limited to a maximum of 500mA.
The board can be supplied by an external, STABILIZED +5V voltage through the V+ terminal as well. For that purpose jumper J1 must be removed first.

I²C master connector

The USB power (Vcc, GND) and the I²C signals (SCA and SCL) are available at a female pin connector. Make sure that the overall current from USB does not exceed 500mA. The I²C pull-up resistors (usually 2K2) must be provided by the user circuitry. Remember that I²C chip address 08hex is occupied by the LTC2309 already.



Software

The software setup copies the following files to your hard-disc:

API
This directory contains files and examples made for some programming languages.

DOC
Contains some useful datasheets.

DRIVER:
This folder contains files necessary for Windows plug & play installation. Connecting the device to the USB these driver files for CH341A chipset installed. The device is then ready for use. In software applications the device is addressed with a device number. This is a consecutive number of available CH341A chips starting from zero. Applications open a data channel to the CH341A chip that will then execute the data transport to the ADC. In most applications the device no. is adjustable.

TEST
A simple test program can be found in this folder.

Functional test

For a simple device test and to find out the device no. use the USB_ADC_TEST.EXE from the \TEST folder.

ProfiLab Expert

This folder contains examples for ProfiLab Expert. Please make sure to have latest update from our Internet site installed.

Programming interface (API)

Programming examples for some languages can be found in the \API directory of the software installation.

The ADC and the I²C interface is programmed using a ActiveX control (file ABACOM_USB_ADC.OCX). The ActiveX control is registered as ”ABACOM USB ADC” during software installation process and can be imported with any programming language that support ActiveX controls. Please refer to the manual of your programming language how ActiveX controls a handled.

The following code snippets are meant as illustration for the functional principles and may fail to compile. Remember the code examples.

To program the board an object ”ABACOM USB ADC” (TUSBADCX) must be created. This could be done dynamically at run-time or with the IDE´s form designer.

```
TUSBADCX *ADC ;

ADC = new TUSBADCX(this);
ADC->Parent = this;
ADC->Left = 24;
ADC->Top = 40;
```

The ADC object defines some properties and methods for general control administration (left, top, etc.) and specialized ones for ADC and I²C control., like

```
Cannel0Value = ADC->ADC_CH0 //single ended read CH0
DiffCH0CH1 = ADC->ADC_DF01  //differential read CH0-CH1
```

or

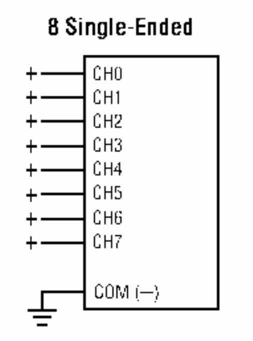
```
ADC->I2C_Write(...)
```

Methods and Properties are named with leading ”ADC_” and ”I2C_”.corresponding to their purpose.

ADC programming

Certain properties (software channels) are read to request voltage values from the ADC.

Single-ended measurement (CH0…CH7; related to COM)

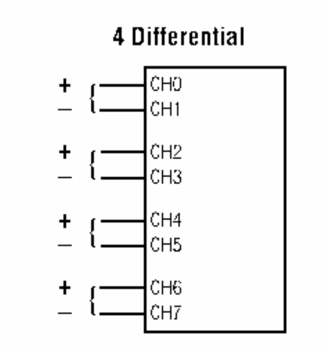


```
   CH0_to_COM_mV = ADC->ADC_CH0 //Read channel 0, single ended
   CH1_to_COM_mV = ADC->ADC_CH1 //Read channel 1, single ended
    (…)
   CH7_to_COM_mV = ADC->ADC_CH7 //Read channel 7, single ended
```

The results of read operations `ADC->ADC_CH0..7` are signed integer values. Between -2047 and +4095. The read result represents the input voltage in millivolts (mV) (with REFCMP = 4,095V).

The LTC2309 provides two possible internal voltage ranges. One of them is unipolar (0…4,095V), the other range is bipolar (-2,047V…+2,047V), both with 1mV/step. To simplify handling the software selects the range automatically. The result is a combined input range from -2047mV to 4095 mV without loss of resolution.

Differential measurement (CH0-CH1; CH2-CH3; CH4-CH5; CH6-CH7)



The terminals CH0..CH7 can be paired as differential inputs as shown in figure above. The results of the differential measurements are read from the software channels 8..11.

```
CH0_minus_CH1_mV = ADC->ADC_DF01 //Read differential CH0-CH1
CH2_minus_CH3_mV = ADC->ADC_DF23 //Read differential CH2-CH3
CH4_minus_CH5_mV = ADC->ADC_DF45 //Read differential CH4-CH5
CH6_minus_CH7_mV = ADC->ADC_DF67 //Read differential CH6-CH7
```

Read result are signed integers, ranging from -4095 mV to +4095 mV. Again the software takes advantage of the LTC2309 internal ranges and therefore reaches a resolution of 1 mV/step, which is theoretically 13 bit.

Connecting signals to the ADC make sure signals are within the „Absolute Maximum Ratings“ of the LTC2309 to prevent damage.

The voltage at CH0…CH7, COM and VREF must not exceed the following limit at any time: (GND) – 0,3V   to   (V+) + 0,3V

This means that input signals must never be negative (if related to PC ground (GND)) and must never exceed the USB supply voltage (V+).

All software channels (single-ended 0..7 and differential 8..11) can be request any time and independently from the actual input circuitry you are using.

In the need to use more than one board, additional ADC objects must be created. To address a device the device number must be set:

```
ADC1->ADC_DeviceNo = 0; //1st device
ADC2->ADC_DeviceNo = 1; //second device
etc.

ADCX->ADC_DeviceNo =-1 //Device disabled (default)
```

I²C (TWI) programming

The device is equipped with an integrated I²C(TWI) master interface, which makes it possible to connect additional peripheral chips and control them with PC software. Some basic knowledge in I²C technology and understanding of chips datasheets is necessary for that.

The programming examples are based on a ready-made I²C demonstration board from MicroChip (TM), which can be purchased separately from our web-shop.
(http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13)

Instead of that, any other I²C circuitry could be connected and programmed.

The following I²C methods are usable for that purpose:

Read/write bytes from/to chip registers

The method

```
   ADC->I2C_Write(0x20,0x09,LEDStatus)
```

Writes a byte (here LEDStatus) to the chip (slave) address 0x20 into a chip register (here 0x09).

```
   ADC->I2C_Read(0x20,0x09,InData)
```

Read a register (here 0x09) of a chip, with chip address 0x20. The return value InData contains a readable ASCII string (HEX dump; e.g. ”A7”)

Direct read/write data over i²C bus

The following stream function allows direct read and/or write over the I²C bus:

```
   OutData = "92 00";
   ADC->I2C_Stream(outData, 2, inData);
```

The parameters (inData; outData) are readable ASCII strings (Hex dumps). The example writes two bytes 0x92 and 0x00 out on the bus. The number of bytes to write is implied with the the length OutData hex dump.
The number of bytes to read must be given in the second parameter (here: two). The return value is a readable hex dump string, like ”FE B6” of the read bytes.

This function writes/reads bytes – without certain protocol – over the I²C bus and is therefore usable with almost any peripheral chip type.

EEPROM

Read/write to EEPROM can be done with

```
   ADC->I2C_WriteEEPROM(eprom24C02, 0, ”FF FE”);
   ADC->I2C_ReadEEPROM(eprom24C02, 0, 7, Data);
```

For write operation the EEPROM type, the memory (start) address and the write data must be fed. For read operations the number of bytes to read must be given in addition (here: seven). Read and write data are readable hex dump strings again.

Clock frequency

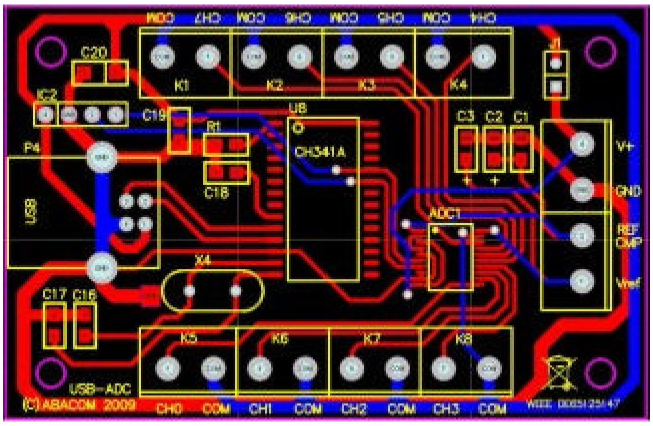
The I²C speed can be adjusted in four steps:

```
      ADC->I2C_Speed = slow_20kHz
      ADC->I2C_Speed = normal_100kHz
      ADC->I2C_Speed = fast_400kHz
      ADC->I2C_Speed = high_750kHz
```

Fast and high clock rates may not be supported by all I²C chips.

Dimensions

70 x 45 x 15 mm


COM
CH7
COM
CH6
COM
CH5
COM
CH4
C20
K1
K2
K3
K4
J1
IC2
C19
U8
R1
C3
C2
C1
P4
CH341A
V+
C18
GND
USB
ADC1
REF
CMP
X4
Vref
C17
C16
K5
K6
K7
K8
USB-ADC
(C) ABACOM 2009
CH0
COM
CH1
COM
CH2
COM
CH3
COM
WEEE DE85125147