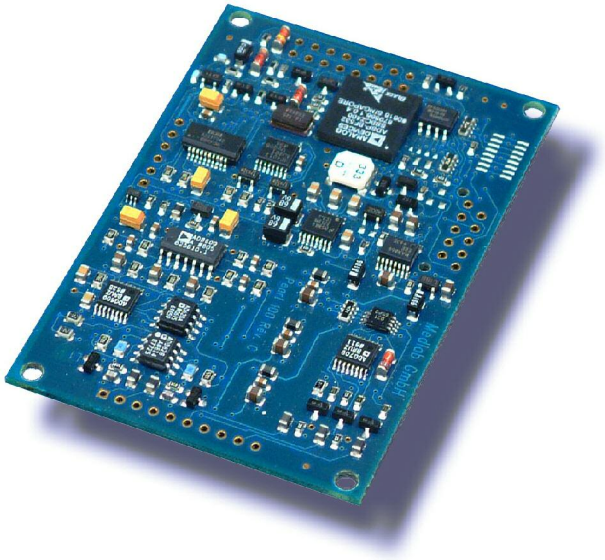
medlab

Pulse Oximeter OEM Board

PEARL100™

Technical Manual



Version 1.46 17.03.2016

PEARL = **P**ulse **E**nhancement **A**rtefact **R**ejection **L**ogic

Revisions:

V1.00	12.07.2006	Added DSUB-JP Adapter drawing
V1.10	10.06.2008	Minor changes, new probe pictures
V1.20	10.10.2010	Added description of perfusion info
V1.30	11.06.2012	changed manufacturer address
V1.40	20.11.2012	Edited sensor connection information
V1.41	21.11.2012	Corrected typing errors
V1.42	11.02.2013	Clearified Ground connections of probe Added mounting instructions page 19
V1.43	29.01.2014	Corrected mounting instructions on page 19
V1.44	28.03.2014	Added description of version transmission, page 12
V1.45	23.10.2014	Added operating temperature
V1.46	16.03.2016	Changed operating voltage range

Medlab medizinische Diagnosegeräte GmbH
Helmholtzstrasse 1, 76297 Stutensee/Karlsruhe, Germany
Tel. +49(0)7244 741100
support@medlab.eu
www.medlab.eu

Contents:

Overview

The scope of this document is the description and specification of Medlab's PEARL100 pulse oximeter module. It is intended to help to integrate the board into another medical device manufacturer's electronic system.

The probes (sensors) that can be used together with the PEARL100 are described on the last pages of this document.

The document describes the basic technology used, the mechanical and power supply considerations and the software protocol for interfacing the PEARL's UART to a host system.

The PEARL is an electronic PCB, fully pre-tested, that connects to one of the different Medlab SpO_2 probes to measure a patient's arterial oxygen saturation and his pulse rate.

The interface to the host system needs a filtered and regulated DC power supply of 3.3 to 5.5 volts and an asynchronous, serial connection to the host system.

Depending on the selected protocols, the baud rate varies between 4800 and 9600 baud, and also depending on protocol, the connection is uni- or bidirectional.

The delivered standard is protocol number 2, this is a unidirectional connection to the host system with a data rate of 9600 baud. Transmitted are the SpO_2 value, the pulse rate, the plethysmographic waveform and several status information.

PEARL Algorithm

The PEARL100 module uses the proprietary PEARL algorithm, that constantly shifts a time window of six seconds over a data buffer that contain samples of the red and infrared waveforms. The algorithm detects the correct pulse rate by convoluting a template over the waveform at different phase angles. This technology is also known as "cross correlating" the signal to another in digital signal processing. On the analog side, the PEARL uses a lock-in amplifier that enables the module to work even under extreme conditions, as high ambient light and other sources of disturbance. The basic operation frequency of that amplifier is 20 kHz, and the used analog front end very effectively suppresses noise by using a lock-in signal recovery scheme.

The PEARL algorithm leads to the effect that real time pulse detection is not any more coupled directly to calculation of correct saturations, since the mentioned six second buffer is used for all calculations.

Since the algorithm itself does not detect the pulse in real time, a traditional pulse detector is applied to the input signal and is responsible for the realtime pulse detection which can be used by the host system to generate a pulse tone.

Once per detected pulse (protocol 2), the current calculated SpO_2 and pulse values are transmitted to the host system.
If an audible signalization is needed, this point in time of data transmission can be used to generate the mentioned pulse tone.

Technical Data (Specifications)

Mechanical Data:

General:	6-layer PCB, thickness 1mm
Attachment:	Four 3.3mm holes in the corners of the PCB
Weight:	23 g

Power Suppy:

Operating Voltage:	3.3 - 5.5 Volt DC, 20...40 mA depending on interface and LED brightness. (LED brightness depends on finger thickness)
Power Consumption:	100mW to 150mW, depending on operating voltage lowest power consumption at 3.3 volts (100mW)

Environmental:

Temperature:	Storage	-30 °C to 90 °C
	Operation	-20 °C to 50 °C
Humidity	Storage	0 .. 95 %, non condensing
	Operation	5 .. 95 %, non condensing

SpO_2 :

Measuring range:	0%..100% of SpO2	
Accuracy:	Standard	+/- 2 %
	Movement artefacts	+/- 3 %
	Low perfusion	+/- 3 %
	(70-100 %, below 70 % not specified)	
Averaging:	Depending on protocol	

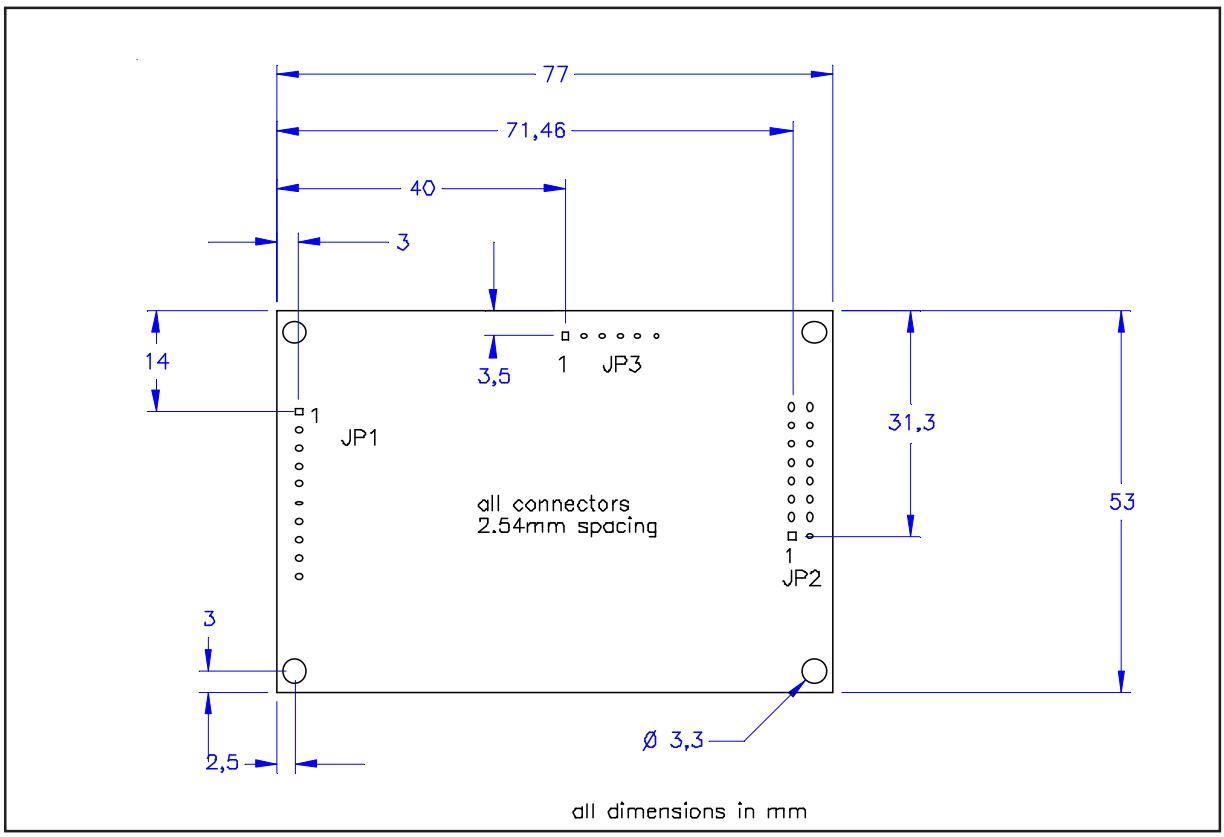
Pulse Rate:

Measuring range:	30 .. 250 min^{-1}
Accuracy:	+/- 3 min^{-1}
Averaging:	Fixed to 8 beats

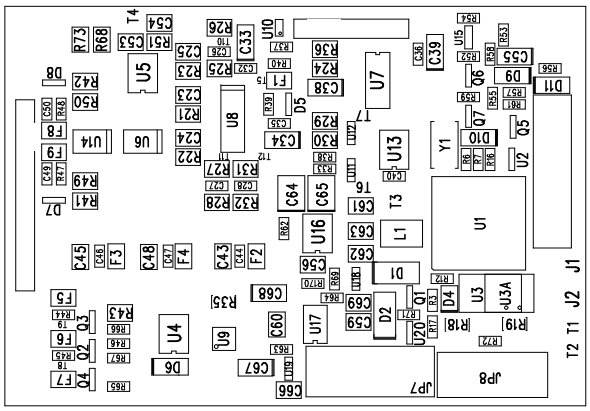
Interface:

Asynchronous, serial interface with TTL or RS232 levels
Baudrate and data format depending on protocol.

Mechanical Dimensions

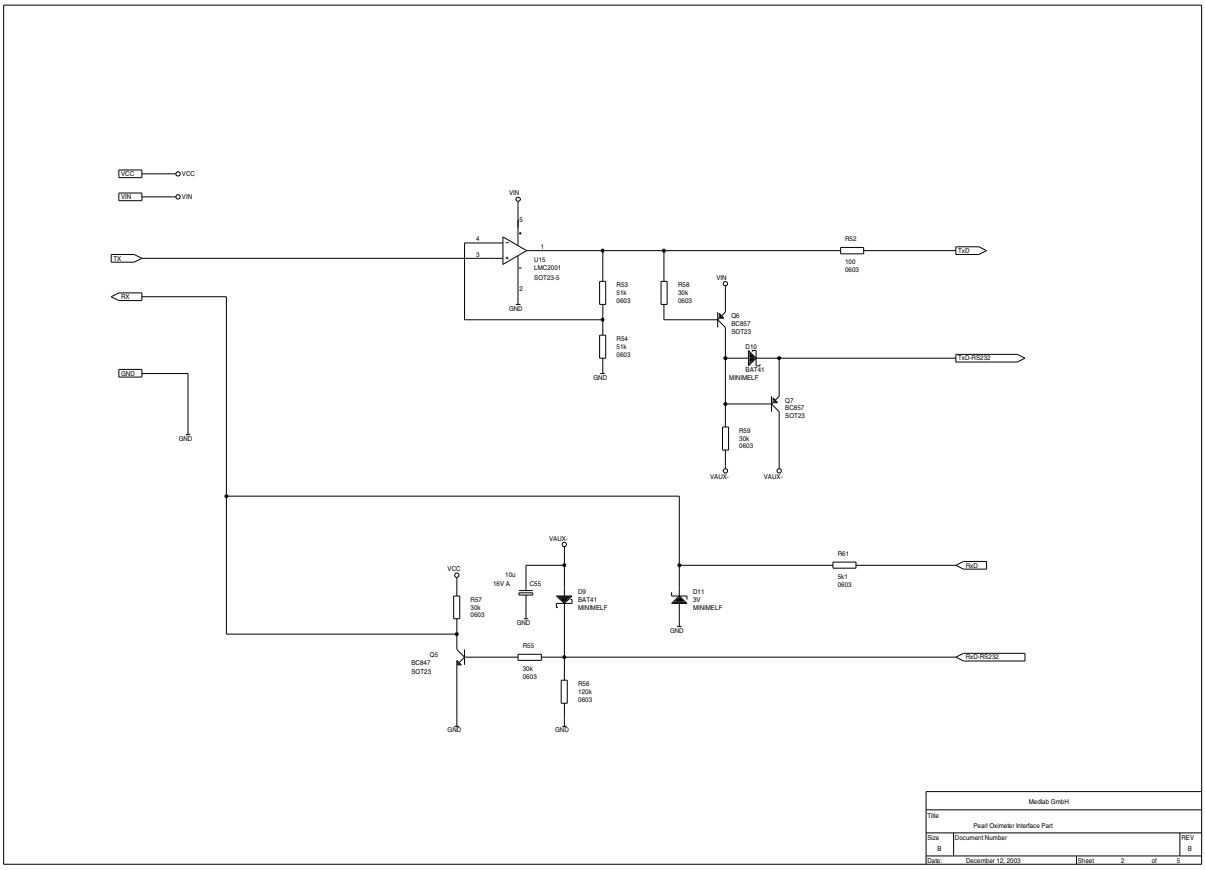


Mechanical drawing of top view of the PCB (original size)



Part position top view of the PCB

Hardware Interface



Interface part of the pulse oximeter´s electronics

Serial Interface

The host system is connected to the pulse oximeter module over a serial, asynchronous communication channel with protocol depending baudrate and transmission parameters.

Both CMOS and RS232 voltage levels are available.

As can be seen on the schematic on top of the screen, the board adapts to the input voltage of the system. That means, if the board is powered by 5 V DC, the RxD, TxD levels are 0 and 5 Volts, +/- 50 mV. If the board is powered by 3.3 V DC, the levels are 0 and 3.3 V, +/- 50 mV.

The RS232 option is in fact not a real RS232 interface, but is helpful during evaluation of the board, which can then be done using an ordinary PC and a special test software. The board uses the positive supply as the "0" output level and uses the negative voltage from the incoming RS232-RxD pin to generate the negative "1" output level. Therefore, RS232-RxD and RS232-TxD must be connected to the host, or no RS232 level output will be available. The board should be powered with 5 V DC when using this option, otherwise, the positive level is not highh enough to trigger the receiving UART reliably.

The connection in the customer's final system should preferably be done through CMOS level connnection pins.

In the two standard protocols, only a unidirectional interface (PEARL100 ---> host system) is necessary.

Power Supply

The PEARL module internally works with a 3.15 V power supply and generates all other voltage used internally from these 3.15 V. The input range has been extended to accept 5 V DC also.

The 3.15 V for the board internally used are generated by a linear voltage regulator that enables the PEARL to work with supply voltages ranging from 3.3 to 5.5 V DC.

Best efficiency is reached when powering the module with 3.3 Volts DC. The 3.3 V should be connected to the two respective pins on JP2. By connecting the voltage to these pins, the internal voltage regulator is disabled and the module is powered directly by the 3.3 V source. It is then important to regulate this supply to an accuracy better than +/- 5%, or permanent damage will occur to the module.

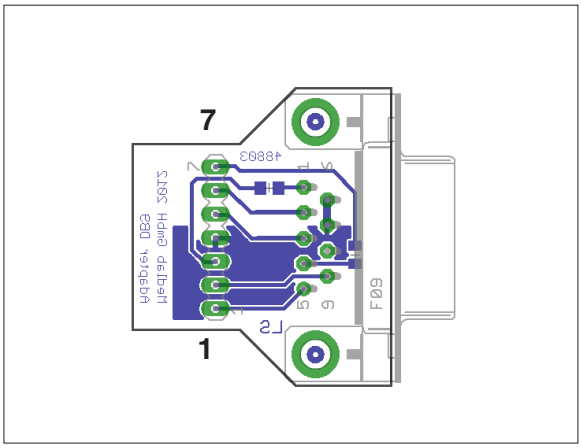
Connectors

(see drawing on page 7)

Header for Probe Connection:

(pin number of probe DSUB connector to connect to in parenthesis)

JP1:	1	Input Photodiode 1 (DSUB 5)
	2	Input Photodiode 2 (DSUB 9)
	3	Sensor Coding (DSUB 1)
	4	Agnd (DSUB 6 and 7)(1)
	5	Led Output 1 (DSUB 3)
	6	Led Output 2 (DSUB 2)
	7	Sdata (DSUB 4)
	8	don't connect
	9	don't connect
	10	don't connect



Adapter for DSUB 9 probe to PEARL100 board

Note: Depending on the RF immunity level your final product needs to fulfil (3V/m or 20V/m), it might be necessary to include further EMC filtering measures close to the SpO_2 connector input of your medical device.
Please contact Medlab for details.

Header for Host Connection:

JP2:	1	Ground
	2	Ground
	3	Txd (CMOS level)
	4	RS232-Txd (RS232 level)(2)
	5	Rxd (CMOS level)
	6	RS232-RxD(RS232 level)(2)
	7	n.c.
	8	n.c.
	9	n.c.
	10	n.c.
	11	/Powerdown
	12	Aux Port
	13	3.3 VDC input
	14	3.3 VDC input
	15	VCC input, 3.3 - 5.5 VDC(3)
	16	VCC input, 3.3-5.5 VDC(3)

Alternative Header for Host Connection :

JP3	1	/Powerdown (Connect to 0V for power down)
	2	RxD (TTL Level)
	3	n.c.
	4	TxD (TTL level)
	5	Ground
	6	VCC input (3.3-5.5VDC)

(1) see page 18, "Mounting the PEARL100 board into your medical device"
(2) see description under "Hardware Interface"
(3) use only pin 13 and 14 **OR** 15 and 16 for supply, do not use both.

Serial Transmission (Protocol 1)

Transmission is not synchronized to any event in the SpO2 calculation, instead a data block is sent 60 times per second. Each data block is five bytes long. The power consumption of the PEARL is slightly higher than in other protocols, due to the relatively large amount of data that is transmitted.

Advantage:
Since the board always transmits 300 bytes per second, it is easily recognized if transmission fails for some reason. Easy to decode
Disadvantage:
The host CPU has to process a relatively high amount of mostly redundant data because data bytes are received 300 times per second.

Data format:
4800 Baud, 1 Startbit, 8 Databits, even parity bit,1 Stopbit

Bit 7 in Byte 1 is used for synchronisation of the blocks.

Byte	Bit 7 6 5 4 3 2 1 0	Definition
1	1	a "1" in bit 7 indicates the start of a new block
	x	pulse trigger (1 for about 100 ms if pulse detected
	x	not used
	x	not used
	x x x x	Signal strength (0..7), 0x0f is bad signal
2	0	
	x x x x x x x	Plethysmogram sample value (0..99)
3	0	
	x	Bit 7 of Pulse
	x	not used
	x	problem with probe
	x x x x	Bargraph 0..15
4	0	
	x x x x x x x	Pulse bit 0 to 6
5	0	
	x x x x x x x	Spo2 (0..99)

This 5 Byte block is transmitted with 60 Hz = 300 bytes/second.

Serial Transmission (Protocol 2)

The second protocol implements the same functionality as the first one, without the large redundant data overhead of it. This protocol is the default protocol when ordering a module from Medlab.

Physical Data format:
9600 baud, 8 bits, 1 stop bit, no parity

Plethysmographic waveform data during measurement is sent at 50 Hz transmission rate. This rate was selected since higher frequencies do not produce smoother waves, lower frequencies are leading to an incorrect appearance of the waveform.

As long as there is no probe connected to the module, or no finger is detected, e.g. the module is issuing info bytes containing either 0x01 or 0x02, no waveform data is transmitted. The marker bytes and pulse and SpO2 values of "0" are sent once per second in this state, the same is true for the info and quality/perfusion bytes with their respective markers.

During measurement, for each detected pulse, a block with new saturation, pulse rate, info and quality/perfusion information is transmitted. The pulse wave sample points are transmitted continuously at 50 bytes per second. Their values are limited to a range between 0x00 and 0xF7, so they do not interfere with the marker bytes. The point in time where the host receives a new data block of markers, SpO2- and pulse values can be used to generate a pulse beep on the host side.

After power up, the module needs about three seconds to finalize its internal boot process. The first three data blocks sent after boot (one per second, if no measurement taking place) contain a firmware revision number in the lower five bits of the quality/perfusion byte, e.g. the byte following the 0xFC marker. As this was introduced with firmware revision 1.28, 128 has been chosen as an offset. To calculate the actual firmware revision, use the following algorithm:

revision = (128+(quality_byte & 0x1F))/100

Note: reading a value of 0x0A from the quality/perfusion byte means you missed the firmware revision, as 0x0A is the default content of this byte, when not measuring.

As mentioned before, values that are higher than 0xF7 are used for marking the following data byte as a new data value with the definitions on the next page:

Marker byte	Definition of following byte
0xF8	wave sample points follow
0xF9	Spo2 value follows
0xFA	Pulse value follows
0xFB	info byte follows
0xFC	quality/perfusion byte follows

Definition of command bytes

Info byte	Definition
0x00	OK
0x01	No sensor connected
0x02	No finger in probe
0x03	Low perfusion
0x45	Selftest Error

Definition of info byte

```
Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0
0    A2   A1   A0   B3   B2   B1   B0

Bit A2 bis A0: Perfusion Info

A2 A1 A0
0  0  0    unused
0  0  1    < 0.25%   AC/DC ratio
0  1  0    0.25-0.5% AC/DC ratio
0  1  1    0.5-1.0%  AC/DC ratio
1  0  0    1.0-2.0%  AC/DC ratio
1  0  1    2.0-4.0%  AC/DC ratio
1  1  0    4.0-8.0%  AC/DC ratio
1  1  1    > 8%      AC/DC ratio
```

```
Bit B3 - B0: Quality

10..0      0 is best quality
```

Definition of quality/ perfusion byte

Perfusion info is a number between 1 and 7, coded into bit 6 to bit 4 of the info byte. It is an indicator for the relation of pulsating - to constant radiation through the measuring site. See table on the left for details.

Quality is a number between 0 and 10, coded into the lower 4 bits of the info byte.
If the number is 0, ten or more consecutive pulses have been detected without artefact or other problems.

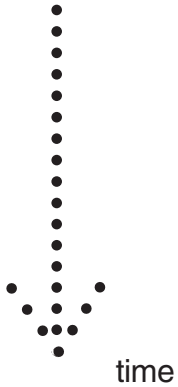
Description of quality/perfusion byte

Examples of Data Transmission

Protocol Version 1:

BLOCK n	
0xC7	*new block, pulse trigger, signal strength = 7*
0x47	*plethysmogramm sample (between 0 and 99)*
0x08	*bargraph sample point is 8, pulse bit 8 = 0*
0x60	*pulse = 96*
0x61	*spo2 = 97*
BLOCK n+1	
0x86	*new block, no pulse trigger, signal strength = 6*
0x49	*plethysmogramm sample (between 0 and 99)*
0x46	*bargraph sample point is 6, pulse bit 8 = 1*
0x30	*pulse = 48 + 128 = 176*
0x60	*spo2 = 96*

.....
.....
.....



Protocol Version 2:

0xF9, 0x61, ...	0xFA, 0x80,	0xFB, 0x00,	0xFC, 0x30,	0xF8, 0x23, 0x25, 0x26, 0x28, 0x30
SpO2 = 97%,	*Pulse = 128,*	*Status = OK,*	*Quality = 0, Perfusion = 3*	*Waveform, sample points follow until next marker byte received*

The advantage of this protocol is the low overhead. The data type with the highest transmission rate, the waveform, is not marked in any form, leading to the smallest possible amount of transmitted data bytes.
If no finger is inserted or no probe is connected, zero is transmitted for both Spo2 and pulse rate, once per second, while the status byte indicates either "no signal" or "no sensor".

C Source Code Examples for Protocol 1

The following C source codes are intended to help integrate the Medlab OEM pulse oximeter board into the customers system. The data is received in the PC's serial interrupt and the values are copied in a data queue that is processed during the main program.

```
void decode_data(void)
        {
          while (!((val = getccb()) & 0x80));    /*  wait for sync bit              */
          if (val & 0x40)
             printf("!Puls!");                   /*  puls trigger active            */

          y = getccb();                          /*  get plethysmogram sample       */

          val = getccb();                        /*  get pulse bar sample           */
          puls_hbit = (val & 0x80)?1:0;          /*  store bit 7 of pulse           */
          bar_graph = val & 0x0F;                /*  store bar_graph value          */

         printf("Puls %03u",0x80*puls_hbit + getccb());
                                                 /*  print pulse                    */

         printf("SpO2 %03u",getccb());           /*  print spo2                     */
       }


/* getccb() returns the next serial value from a queue that gets filled during the PC's
serial interrupt */
```

C example for decoding of data protocol 1

C Source Code Examples for Protocol 2

The following C source code is intended to help integrate the Medlab OEM pulse oximeter board into the customers system. The first example is part of the sourcecode we used for writing our PC demo program and is written in C. The second example was originally written for usage with a 8051-family controller using the Keil C51 compiler. The data is received in serial interrupt and the values are copied into global variables that can be processed during the main program.

```
/* getccb() returns the next serial value from a queue that is filled during the PC's serial interrupt */

 while (1)
  {
   if ((val=getccb()) == 0xF8)
     {
      while((val=getccb()) < 0xF0)
          {
                    /* here "val" always contains a new plethysmogram sample          */
                    /* process it acccording to your needs .....                      */
          }
     }

   switch(val)       /* now val contains a marker, indicates next byte is a special value   */
     {
       case 0xF9:
                 printf("%02u",getccb());                    /* print SpO2              */
          break;
       case 0xFA:
                 printf("%03u",(unsigned char)getccb());     /* print pulse             */
                 break;
       case 0xFB:
                  switch(getccb())
                  {
                   case 0: gotoxy(20,23);
                           printf("          OK !        ");  /* print messages          */
                           break;
                   case 1: gotoxy(20,23);
                           printf("    No sensor connected !    ");
                           break;
                   case 2: gotoxy(20,23);
                           printf("    No finger in probe !     ");
                           break;
                   case 3: gotoxy(20,23);
                           printf("    Low perfusion     !      ");
                           break;
                  }
                 break;
       case 0xFC:
                 val = getccb();
                 printf("%02u",getccb()&0x0F);                 /* print quality, mask perf.*/
                 break;
     }
  }
```

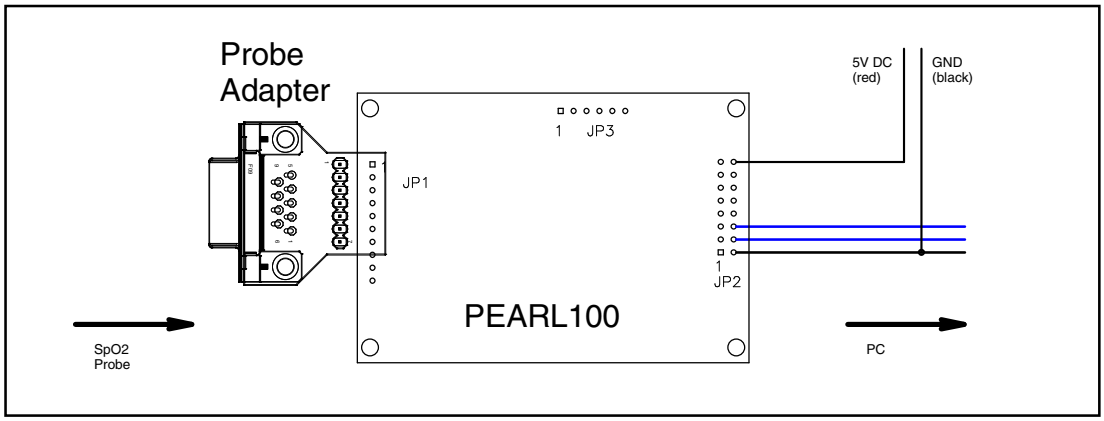
C Example for PC decoding of protocol 2

```
data byte data *rcvptr;
data byte Oxval;
data byte Oxgraph;
data byte Oxpuls;
data byte Oxinfo;
data byte Oxqual;

data bit  Tbit;
data byte Serval;

void serial_int() interrupt 4 using 2
{

 if (TI)                                  /* transmitter int ?                  */
   {
    TI = 0;
    Tbit = TRUE;
    return;                               /* nothing to do                      */
   }

 RI = 0;                                  /* else must be receiver int          */

  Serval = SBUF;                          /* get value from serial buffer register */
   if (Serval > 0xF5)                     /* is it a code ?                     */
     {
      switch (Serval)                     /* yes                                */
        {

         case 0xF8:   rcvptr = &Oxgraph;  /* next time get plethysmogram        */
                      return;
         case 0xF9:  rcvptr = &Oxval;     /* next byte is get spo2 value        */
                      return;
         case 0xFA:    rcvptr = &Oxpuls;  /* next byte is puls value             */
                      return;
         case 0xFB:    rcvptr = &Oxinfo;  /* next byte is spo2 info             */
                    return;
         case 0xFC: rcvptr = &Oxqual;     /* next byte is  quality information  */
                    return;
         default  :    return;
        }
      }
    else
       *rcvptr = Serval;                  /* byte is no code, so store it where pointer points */
   }
 return;
}
```

Code for interrupt driven decoding of protocol 2 using an 8051 microcontroller

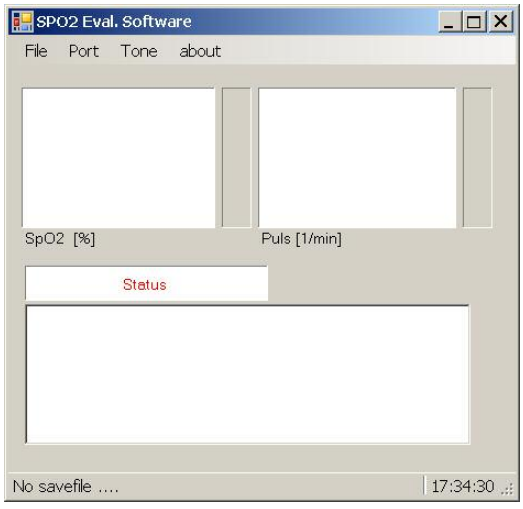
Testkit

To ease evaluation of the module, there is a complete, ready-to-run testkit available: it includes a PC software that is adapted to the pulse oximeter interface protocol. The software displays all relevant data that is transmitted in the protocol version. It runs on each PC under Windows. Also a complete set of cables and adapters is included in this kit. It contains one fingerclip probe P-200 and an adapter to connect the probe to the PEARL100 board



Connection of the Pulse oximeter to probe and PC adapter

Usage



Connect the red cable to a regulated 5VDC voltage source and the black cable to the power supplies GND.

Connect the serial cable to a COM port of the computer. Only Ground, TxD and RxD are used in the interface. If you use a USB-RS232 adapter, please install the "virtual serial port" driver delivered with the adapter.

Connect the other side of the cable to the PCB as shown in the drawing

Connect the probe as shown in the drawing, using the adapter

Turn on the power supply

Turn on PC

Start the software

Regulatory Considerations

The PEARL100 module described in this document is not a final medical product. That means that it cannot be used as a stand alone unit to do pulse oximetry measurements on patients. Therefore, the module does not have to be - and cannot be CE-marked. The customer has to undertake the procedure of CE-marking the final product that will contain the PEARL100 module.

One important part of this is EMC testing the complete medical device. Thsi has to be done on the complete product, not with the board alone. It is important to supply the PEARL100 with a properly filtered, stable DC voltage. The probe inputs and outputs are already fed through small ferrite beads on the module itself.

The module complies to the following standards :

IEC60601-1:1996 and 2006
ISO9919:2005 and ISO80601-2-61

If the final medical device is mains powered or the device can be connected to another mains powered part while a patient is connected, an isolation is required, because ISO9919 requires the patient connection of a pulse oximeter to be of type BF or CF. That means that the power supply and the communication channels for the PEARL100 module have to be galvanically isolated from the main electric part of the medical device. We can support our customers with the design of a simple, reliable and cost effective solution for this.

Mounting the PEARL100 into your Medical Device

If the PEARL100 board is mounted as a piggy back on your own electronic PCB, please make sure that the "Ground" potential of your device is connected to JP2 only, and not also to pin 4 of JP1, the probe connector. Pin 4 of the probe connnector should be connected to pin 6 and 7 of the DSUB probe connector only.

Although on the board itself, these two pins (GND and AGND) are connected at a single point, connecting of pin 4 of JP1 also to your device's ground potential would form a ground loop, that largely deteriorates RF immunity of the PEARL100.

Pulse oximeters used during transport are subject to a 20V/m RF immunity test, instead of the 3V/m used for normal pulse oximeters. In this case, addtional EMC filtering is needed, close to the DSUB connector. Please contact Medlab for details.

Available Probes

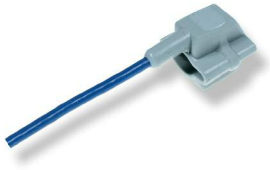
P 200
Fingerclip probe

WR 200
Wrap probe

PS 200
Small Fingerclip probe

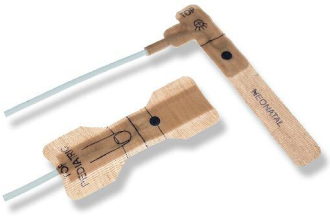
Y 200
Universal Y-probe

R 200
Oxiflex large finger probe



RS 200
Oxiflex small finger probe

V 200
Tongue probe - for
veterinarian applications



Disposable probes
Delivery:
Package of 24 pieces

This page left blank intentionally

Medlab medizinische Diagnosegeräte GmbH
Helmholtzstrasse 1, 76297 Stutensee/Karlsruhe, Germany
Tel. +49(0)7244 741100
support@medlab.eu
www.medlab.eu