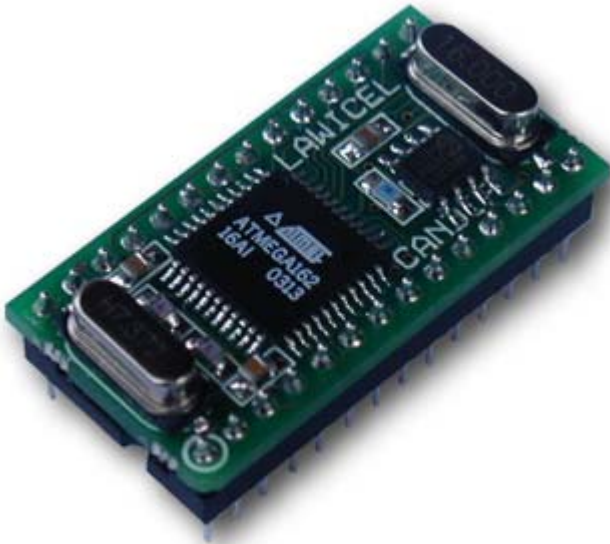


Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

DATA SHEET



CANDIP/M162

Single Board Computer in DIP28 format with
Atmel® AVR® ATmega162 microcontroller and
Philips® SJA1000 CAN 2.0B controller.

Product specification

August 28 2003

CANDIP/M162 Datasheet

Version 1.3

History:

Version 1.0 / Jan 2000 / Preliminary Specification
Version 1.1 / May 2000 / Product Specification
Version 1.2 / September 2002 / Product update
Version 1.3 / August 2003 / Product update to Mega162

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

TABLE OF CONTENTS

Description	3
What can I do with CANDIP/M162	4
Features	4
CANDIP/M162 Software	6
CANDIP/M162 Activity Board	8
CANDIP/M162 Starterkits	9
Physical dimensions	9
CANDIP/M162 Products	9
Company Information	10
Notes	11

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

DESCRIPTION

CANDIP/M162 is an embedded microcontroller with CAN (Controller Area Network), where you as a user can make a quick route to new products, maybe you just need some digital I/O connected to CAN or a smart node with a PID controlled motor etc. You can see the CANDIP/M162 as a very small "Single Board Computer" with the possibility to perform almost 8MIPS.

CANDIP/M162 isn't just a "ready to use module with pre-programmed software" which has a very narrow application field, such as a CANopen SLIO or a homebrewed Protocol SLIO, it is completely scalable with your own software (written in C, Basic, Forth, Pascal or any other compiler that supports the Atmel AVR ATmega162). We can of course pre-program it to behave as you like in case you do not have the time or knowledge.

CANDIP/M162 also has on-board RS-232 transceivers and a RESET circuit that generates a proper reset after power up and will also halt the microcontroller if the voltage drops below 4.38V. Furthermore it has a standard ISO-11898 CAN transceiver on board, but could also use an external CAN transceiver with your own hardware with e.g. OPTO isolation.

CANDIP/M162 will get you a quick start where you do not have to start from scratch and debug hardware while trying to get your application running. We also provide simple routines for testing the CAN controller by sending and receiving frames. Today we provide samples in **ImageCraft ICCAVR**, **CodeVision AVR**, **IAR**, **GCC** and **MCS Electronics BASCOM-AVR**. It is also possible to buy a full working CANlib for the SJA1000, written in ICCAVR or CodeVision AVR with full source code (or as binary LIB files at a lower cost).

CANDIP/M162 has 13(or 15 without RS232) I/O connections on board that could be used for general I/O, but it can easily be expanded to a real advanced and sophisticated system through the SPI port. Through the SPI port you could add hardware such as digital I/O, ADC, DAC, RTC, EEPROM, LCD and much more. You can also attach I2C devices and control them by software controlled I2C. There is also two external Interrupt lines available (INT1 & INT2), the other on the ATmega162 (INT0) is used for the CAN controller.

CANDIP/M162 support In-System-Programming (ISP), which means you can program the software without removing it from the socket. Just use your favorite AVR ISP programmer or buy one from us with the CANDIP (we make a low cost for parallel port called ispAVRp).

We can offer a complete solution and get you started in just a few hours (such as compilers and customized software).

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

What can I do with CANDIP/M162?

- CAN SLIO with your own protocol or with e.g. CANopen, DeviceNet, SDS, CAN Kingdom etc.
- Smart nodes with PID algorithms for controlling e.g. temperature, motors or valves.
- Data Acquisition via CAN. Add your favorite SPI/I2C ADC and collect data.
- Home Automation. Build smart nodes for turning ON/OFF lights, burglar systems etc.
- Add CAN simple in an existing system via e.g. SPI, RS-232 or your own way.
- Education. CANDIP offers a very low cost to learn CAN, we have many starter kits options.
- If you have a customer who want to see a prototype or even a ready product next week.
- Decentralized smart nodes for e.g. Robots.
- CAN monitor in an existing system. Add some Leds and you could monitor e.g. valves etc.

Features

- Standard 28 pin DIP board with 0.1" pins, so use a standard DIP28 carrier.
- Needs only a 5VDC/30mA power source.
- Atmel AVR type ATmega162 working at 7.3728Mhz (other crystals at request).
- 16Kbytes user FLASH, 1Kbytes user RAM and 512bytes user EEPROM.
- Up to 13 digital I/O points on DIP28 board, each capable to sink 20mA as outputs.
- SPI port for expansion.
- Two Interrupt lines available for user functions (INT1 & INT2).
- SJA1000 CAN controller working at 16Mhz, which supports CAN2.0B.
- 82C251 High Speed CAN transceiver 1Mbit (ISO-11898).
- CAN controller can be interrupt driven (INT0).
- Sipex SP202 RS-232 transceiver which together with the AVR can send/receive up to 115kbit, 15KV isolated!
- MAX825M Reset circuit, the normal RESET and inverted RESET is on external pins.
- No interpreted software, it's programmed with compilers.
- Possibilities to implement higher level protocols, such as CANopen, DeviceNet etc.

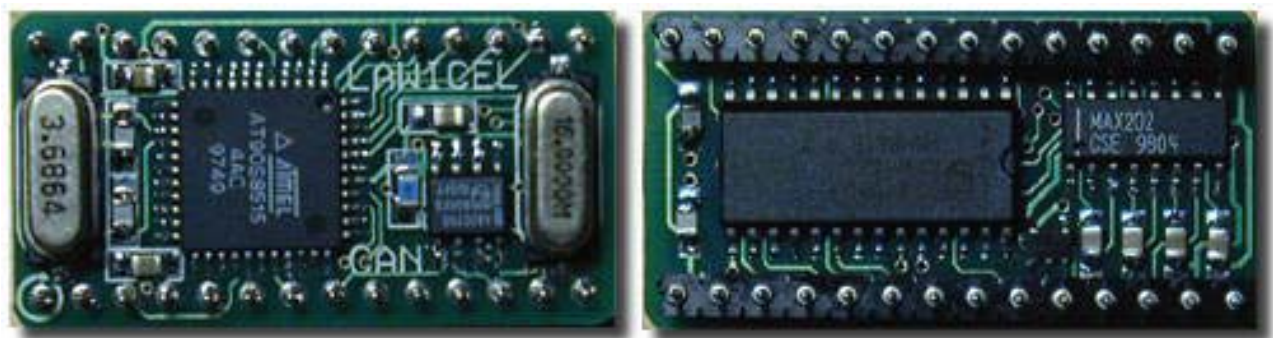


Figure 1: shows the CANDIP/M162 board (both sides) mounted with an ATmega162, SP202, MAX825, SJA1000 and a 82C251 CAN transceiver. You only need to apply power and connectors of your choice and you are up running. No hassle trouble shooting hardware while learning CAN. Note, picture shows the original CANDIP/AVR1.

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

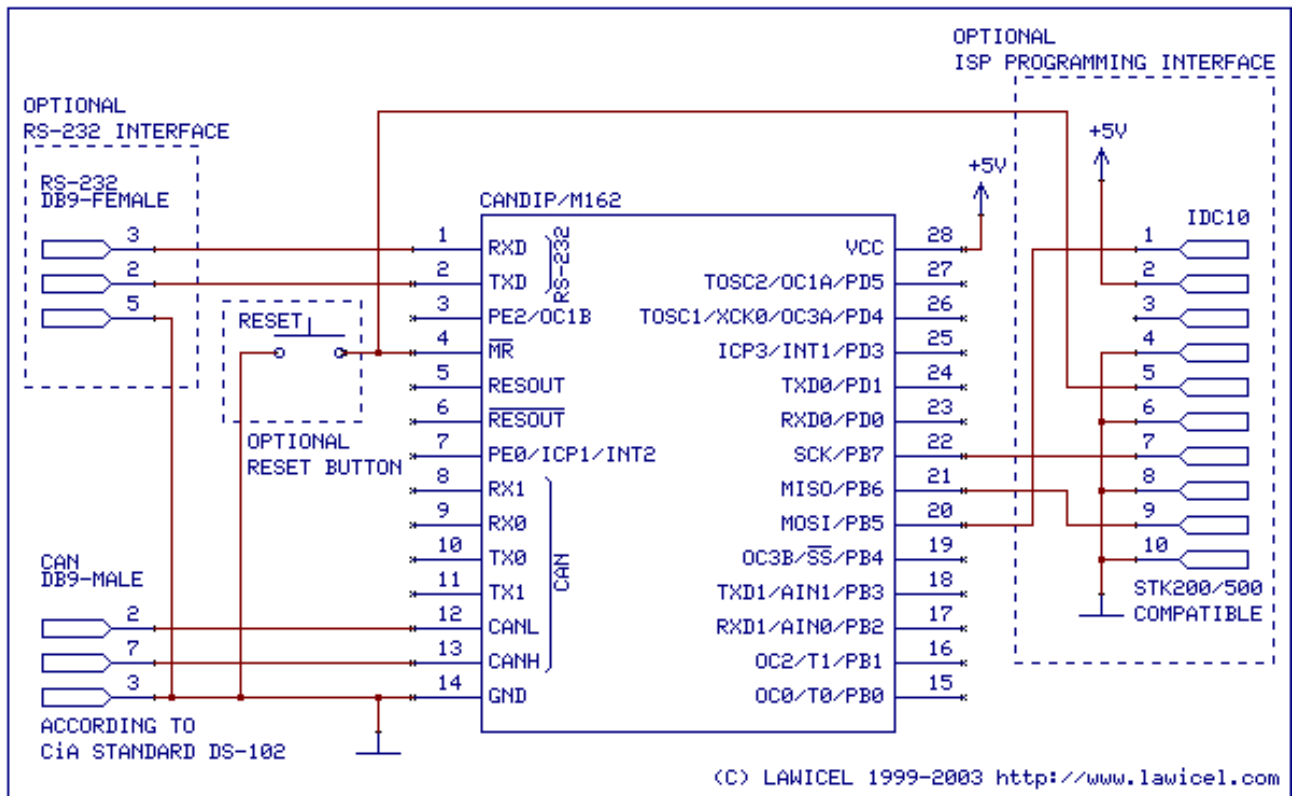


Figure 2: shows a block diagram of CANDIP/M162 and external components needed. Dashed blocks are optional connections depending on your application. The CANDIP/M162 supports In-System-Programming (ISP) through the SPI port and it supports the STK200/500 programmer from Atmel (and other 3:rd party programmers).

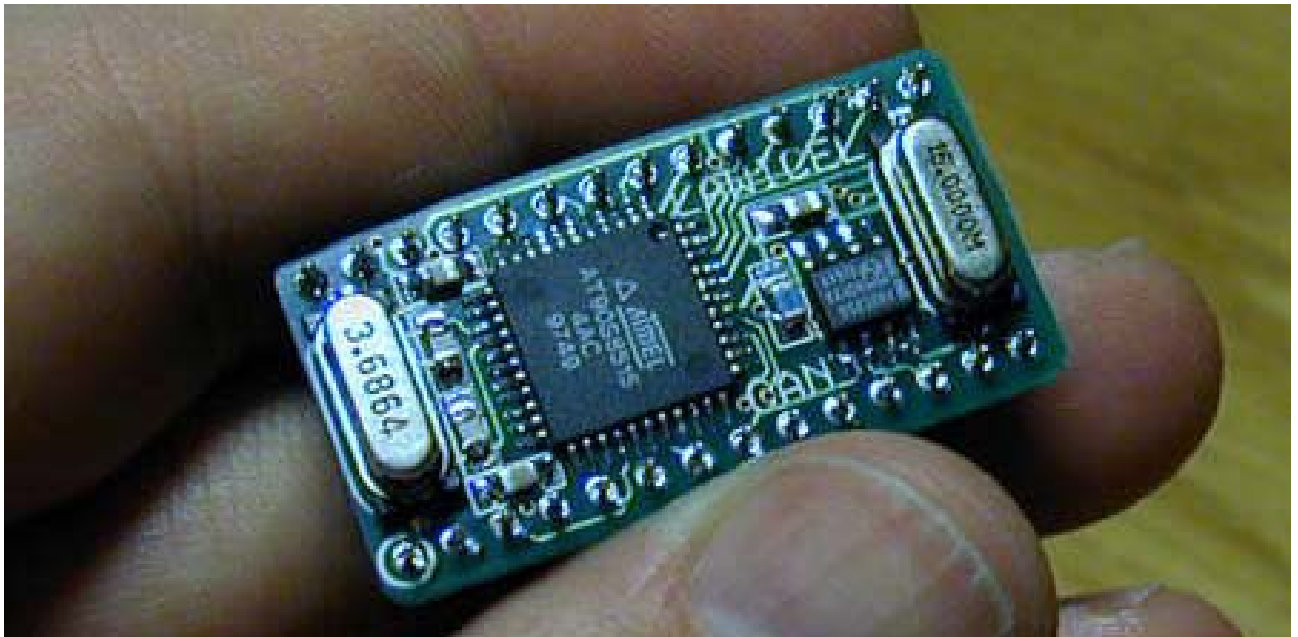


Figure 3: shows how small the CANDIP/M162 is. CANDIP/M162 is perfect for small systems such as Home Automation, small and effective CAN nodes with e.g. digital I/O or Analog I/O, use it for expanding your existing embedded system through the SPI or RS-232 channel. It can also be used for simple test and learning CAN. The CANDIP/M162 is easy to expand into a large system through the SPI port or via software controlled I2C.

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

CANDIP/M162 Software

WiCAN CANlib for ImageCraft ICCAVR

WiCAN CANlib is an easy way of getting up running quickly and you do not need a great knowledge in how CAN works. WiCAN supports both 11bit and 29bit Identifiers in Philips PeliCAN mode, easy initialization of the CAN controller with standard bitrates as well as user defined, FIFO queues for receiving and transmitting CANframes through interrupt control, status information etc. The reason for minimal support is to minimize the resources in the AVR, so there is plenty of room for the actual application both in RAM and FLASH. It is easy to change the support to extended frames, but since many HLP (Higher Level Protocols) and applications today do not use or support the extended frame we didn't include it as default, just to save resources.

There are 3 versions of the WiCAN CANlib for ICCAVR & CodeVision AVR. A free version (demo) with fixed 20kbit bus speed and FIFO queues are preset to 2 CAN frames. With this version it is possible to test the CANlib and to see if it fits your needs. Then next version (CLIB/1) is in Object form (LIB) and cannot be altered or adjusted, but there is no limits in the speed. The last version (CLIB/2) is with full documented source code and here is possibilities for you to modify or add functions to suit your need. The source code is treated as a site licence and you may not distribute or sell the source code further on. Contact us for more information regarding this.

CLIB/1 is on Object form and can not be changed. Both FIFO queues are preset to 4 CAN frames. This is an inexpensive way of testing the CANlib with full functionality and speed. All you need to do is to include the LIB in the project and add a header file where all CANlib definitions are done. Sample programs are included to show how it works. We can of course change the size of the FIFO queues to suit your needs.

CLIB/2 is in fully documented C source code and be changed to suit your need. Both FIFO queues can be set to suit your needs but only in size of 1,2,4,8,16... to maximum RAM of the AVR. All you need to do is to include the WiCAN.C and WiCAN.H file to your project and you are up running. Sample programs are included to show how it works.

A simple C source code to demonstrate how to setup the CAN controller, send a frame and receive some frames, detect the received frames and then transmit them back on the bus can be seen on next page.

The WiCAN CLIB demo can be downloaded from <http://www.candip.com>

Single Board Computer in DIP28 format with Atmel® AVR®
 ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

```

/*
** File:      Main.C
**
** Purpose:   Demonstrate WiCAN LIB for SJA1000
**
** Chip:      ATmega162 running at 7.3728
**
** Version:   1.0.5, 1:st of July 2003
**
** Author:    Lars Wictorsson
**            LAWICEL / SWEDEN
**            http://www.lawicel.com   lars@lawicel.com
**
** Copyright: The copyright to the computer program(s) herein is the
**            property of LAWICEL HB, Sweden. The program(s) may be used
**            and/or copied only with the written permission of LAWICEL HB
**            in accordance with the terms and conditions stipulated in
**            the agreement/contract under which the program(s) have been
**            supplied.
**
** Remarks:   This program is tested with ICCAVR version 6.28.
**
** History:   2000-05-18  1.0.0  Created (LWI)
**            2001-05-06  1.0.1  Modified to test 29bit (LWI)
**            2001-05-14  1.0.2  Modified to test RTR (LWI)
**            2002-01-23  1.0.3  Re-Compiled with latest ICCAVR (LWI)
**            2002-08-16  1.0.4  Re-Compiled with latest ICCAVR & WiCAN 1.0.6 (LWI)
**            2003-07-01  1.0.5. Re-Compiled with ICCAVR 6.28, ATmega162 & WiCAN 1.0.7 (LWI)
*/

#include <iom162v.h>
#include <stdio.h>
#include <macros.h>
#include "WiCAN.h"

WiCAN_Object MyFrame;

void main( void)
{
    UBRR0 = 7;                // 57600 baud for 7.3728 MHz clock
    UCSR0B = 0x18;           // Enable RXEN & TXEN
    /*
    ** Setup CAN controller
    */
    if ((WiCAN_Init(WiCAN_SPEED_20K, 0, 0, 0xDE) == WiCAN_OK) {
        WiCAN_Start();

        MyFrame.id      = 0x500;           // Set up a standard 11-bit ID Frame with
        MyFrame.len     = 1;              // DLC = 1 (i.e. 1 byte)
        MyFrame.byte[0] = 0x01;           // The first byte to '0x01'
        WiCAN_SendFrame(&MyFrame);        // Send the frame

        MyFrame.id      = 0x510;           // Set new ID
        MyFrame.flags    = WiCAN_29;       // Change the frame type to Extended
        MyFrame.byte[0]++;                 // Increase the value of byte 1
        WiCAN_SendFrame(&MyFrame);        // Send the frame again

        MyFrame.id      = 0x520;           // Set new ID
        MyFrame.flags    = WiCAN_RTR;      // Change the STD frame type and set RTR bit
        WiCAN_SendFrame(&MyFrame);        // Send the frame again (notice DLC is set to zero when sending)

        while(1) {
            // Do a forever loop
            if (WiCAN_GetRxQueueSize() >= 4) { // Wait until input FIFO has 4 messages or more
                while (WiCAN_GetFrame(&MyFrame) == WiCAN_OK) {
                    WiCAN_SendFrame(&MyFrame); // Send all of them back as they came in.
                }
            }
        }
    }
}

```

Figure 4: C source code demonstrating the WiCAN LIB.

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

CANDIP/M162 Activity Board

This is the Activity board (ACB1) for CANDIP/M162. With this board you get a quick start prototyping and testing CAN and your own system. This board is great for educational purposes and a low cost alternative to wire up something on your own.



Figure 5: Shows the Activity Board for CANDIP.

ACB1 Features:

- Board covered with a ground layer on component side for good EMI performance.
- 28 pin DIP carrier for the CANDIP/M162.
- AC/DC power input in standard 2.1mm, not sensitive to polarity. 7-15V AC/DC. A 7805 that generates +5VDC and a green LED for +5V OK indication.
- ISP connector for In system programming (STK200 compatible) with red led for programming status indication.
- Reset button.
- DB9 female for easy RS-232 connections.
- DB9 male for CAN (CiA standard DS102). Possible to feed other nodes via one power supply and the CAN cable (selectable via jumper).
- CAN Termination resistor (120ohm) selectable via jumper.
- 3 ea. jumpers for input, can work as Node addressing (0-7) or other settings.
- 4 buttons and 4 leds (all 4 buttons and leds cannot work at the same time, but it is possible via jumpers to select e.g. 3 leds and 1 button).
- Large prototyping area with all signals from CANDIP/M162 easy accessible from the ends including many for power and ground. SPI signals from CANDIP/M162 is not hooked up with LED's or buttons, so it easy to expand the I/O points with SPI slave devices.

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

CANDIP/M162 Starterkits

CANDIP/ST1M162 consist of one CANDIP/M162, one Activity board ACB1 (no sample programs included). As an option it is possible to include compilers, programmers, AC/DC adapters cables etc. Just ask us what you need and we can offer a bundle price.

CANDIP/ST2M162 consists of two CANDIP/ST1M162 plus one RS-232 cable, one 1 m. CAN cable, AC/DC adapter (not included in all countries) and sample programs demonstrating distributed I/O and the Philips PeliCAN mode. When ordering CANDIP/ST2M162, we offer you a 10% discount on ImageCraft C compiler and MCS Electronics BASCOM-AVR compiler (this offer is only available if ordered direct from LAWICEL, check with your local dealer if they have similar offers). Sample programs are provided in ICCAVR, CodevisionAVR, GCC, IAR & BASCOM.

Physical Dimensions

The CANDIP/M162 has the same footprint as a standard DIL28/DIP28 IC circuit and thus fits perfectly in any DIL28 IC carrier (it can also be soldered directly into the holes of the motherboard). The pins are 0.5mm in diameter. The actual board dimensions are 19 mm wide and 38 mm long centered around the DIP28 pin layout. Distances between pins on one side are for a standard DIP28 0.1" (or 2,54mm) and 0.6" (or 15,24mm) between the 2 set of 14 pins. The CANDIP/M162 is 15 mm high from the edge of the pin to the top of the chrystal. However when e.g. solder directly to the motherboard the pins goes into the holes and the high is about 11,60 mm meassuring from the Printed circuit board to the top of the chrystal.

CANDIP/M162 Products

Product	Remarks
CANDIP/M162	CANDIP/M162 fully mounted with RS232 tranceiver, enables 13 I/O.
ACB1	Activity Board for CANDIP/M162, fully mounted.
CANDIP/ST1M162	CANDIP/M162 Starterkit (1 each of CANDIP/M162 and ACB1).
CANDIP/ST2M162	CANDIP/M162 Starterkit (2 each of CANDIP/M162 and ACB1), RS232 cable, CAN cable, AC/DC adapter* and demos).
CLIB/1	CAN LIB in OBJ for ICCAVR C or CodeVision (specify compiler).
CLIB/2	CAN LIB in SOURCE for ICCAVR C or CodeVision.

* Not included in all countries

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

Other products and companies referred to herein are trademarks or registered trademarks of their respective companies or mark holders.



<http://www.lawicel.com>

<http://www.candip.com>

DEFINITIONS

Data Sheet Identification	Product Status	Definition
Preliminary Specification	<i>Beta version / in Design</i>	This data sheet contains preliminary data and more data will be published later. Specifications may change without prior notice.
Product Specification	Full Production	This data sheet contains the Final Specification. LAWICEL reserves the right to make changes at any time without prior notice, in order to improve the design.

LAWICEL reserve the right to make changes, without notice, in the products, or/and software, described or contained herein in order to improve design or/and performance. LAWICEL assumes no responsibility or liability for the use of any of these products.

LAWICEL HB
Klubbgatan 3
S-282 32 Tyringe
SWEDEN
Telephone +46 451 59877
FAX +46 451 59878
<http://www.lawicel.com>
info@lawicel.com
<http://www.candip.com>
info@candip.com

© Copyright LAWICEL 2003
 All rights reserved. Printed in Sweden.

Single Board Computer in DIP28 format with Atmel® AVR®
ATmega162 microcontroller and Philips® SJA1000 CAN2.0B controller.

CANDIP/M162

NOTES