

Crumb128-CAN

Rapid Prototyping Module with AT90CAN128 μ Controller.

Crumb128-CAN is a low-cost, easy to use and small-formfactor module combining Atmels AT90CAN128 AVR microcontroller with integrated CAN controller with a standard serial port with RS232 transceiver, USB2.0 (full speed) device interface, reset protection circuitry, reset jumper, status LED, standard 6-pin InSystemProgramming (ISP) header and all AT90CAN128 signals on two 32pin headers.



Connector Kit – Crumb128-CAN is being shipped without pin headers mounted, since everybody has it's own favorites (pins male/female facing up/down or 90° angled, etc.). A set of high quality pin headers and receptacles are available as a Crumb128-CAN Connector Kit (see picture).

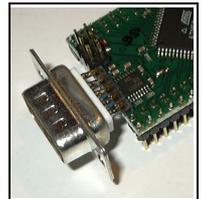
Power Supply – Crumb128-CAN requires a regulated +5V DC power supply. See pinout diagram for VCC pins (both VCC pins are internally connected). The AVCC pin is connected to VCC through coil L1 and decoupled by capacitor C5, hence it should not be connected externally.

System Reset – Reset circuitry C11, R7 and D3 provides a proper reset signal after power up and provide protection against irradiation. By closing jumper J1 a manual reset can be triggered.

System Clock – The CPU clock is generated by a 16.0000MHz crystal (X1) for error free common serial baudrates. An additional 32.768kHz asynchronous timer crystal can be used for a software realtime clock.

CAN Controller – The AT90CAN128 provides an integrated CAN controller 2.0A and 2.0B with 15 full message objects with separate identifier tags and masks, transmit, receive, automatic reply and frame buffer receive modes, 1Mbits/s maximum transfer rate at 8 MHz and time stamping, TTC & listening mode (psying or autobaud). The CAN interface signals are available at pins 14 and 15 of the modules headers, see section Connectors below. A sample schematic how to connect a PCA82C250 CAN transceiver is available on the homepage.

Serial Port – RS232 level signals are provided by the onboard MAX3221 (U2) RS232 transceiver, connected to UART0 of the AT90CAN128. U2 can be enabled/disabled by jumper J2, which should be set to 1-2 (disabled) or 2-3 (enabled) position before operation. The RS232 signals are available at CON4, which can be connected 1:1 to a male Sub-D-9 connector (see picture).



USB Port – The USB interface of Crumb128-CAN is based on Silicon Laboratories CP2102 USB-UART converter, which translates converts the USB data flow into a serial UART format. The CP2102 data sheet, as well as royalty-free drivers for Windows, MacOSX and Linux are available on <http://www.chip45.com>. Please install the USB driver prior to connecting the module to the PC.

Furthermore the CP2102 offers an integrated EEPROM for storing USB parameters like VID, PID, product string and serial number, hence providing the possibility to individualize a product based on Crumb128-CAN, by customizing these parameters. Silabs offers a free Windows tool, as well as another tool for generating a customized Windows USB driver. By providing such a driver to your customers, your product will be recognized immediately when connected to a PC and will be registered in Windows control panel with it's unique product name. Later your product will be assigned the same COM port number each time it is connected to the PC, which simplifies access to your product from PC applications.

The tools and the corresponding application notes (AN144 and AN220) by Silabs are also available on <http://www.chip45.com>.

Status LED – LED1 is hardwired to pin 17 (OC2 OC1C PB7) of the AT90CAN128 and can be used by the application as a general purpose status indicator. The bootloader flashes the LED three times after power up.

In-System Programming (ISP) – CON3 is the standard 6-pin Atmel AVR ISP connector and can be used with most ISP adapters (e.g. CrispAVR-200 or CrispAVR-USB by chip45.com or Atmel's stk500) and software (e.g. uisp, avrdude, PonyProg, etc.). Keep in mind, that the AT90CAN128 shares the ISP signals PDI/PDO with UART0, hence serial communication might be blocked as long as the ISP adapter is connected.

Memories – The AT90CAN128 provides 128kbytes of onchip, non-volatile Flash memory for program code storage, 4kbytes of onchip application SRAM and additional 4kbytes of onchip non-volatile EEPROM memory.

If your application requires more memory, CrumbMEM provides 64kbytes of external SRAM memory. CrumbMEM is designed as a pincompatible expansion module and can be stacked underneath Crumb128-CAN, hence providing a very space saving solution.

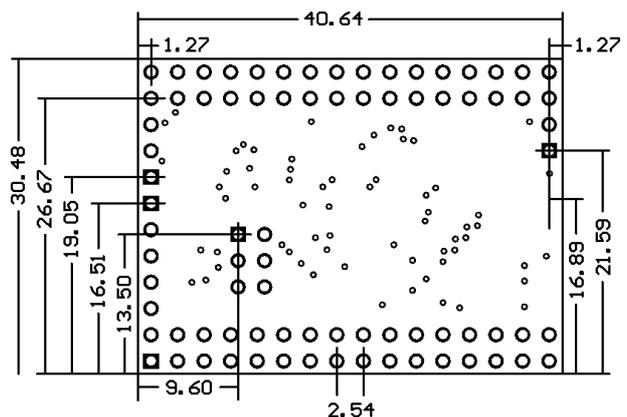
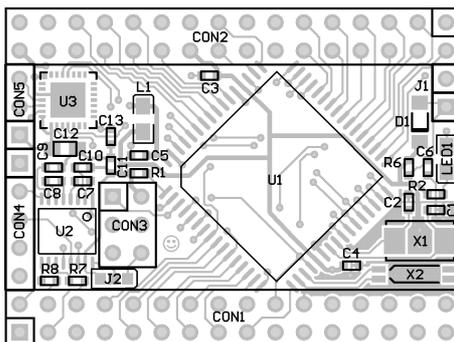
Fuse Bits – The AT90CAN128 fuse bits are not preset to any specific values. If you want to use the 16.000MHz clock crystal instead of the internal oscillator (which is default) set CKSEL[0..3] fuse bits to 0xF.

Connectors - All connectors have standard 2.54mm spacing. CON1/2 provide all AT90CAN128 signals, CON3 is the ISP header, CON4 and CON5 provide the RS232 and USB interface signals.

CON1		CON2		CON3							
NC	<input type="checkbox"/>	TXD0 PDI PE0	<input type="checkbox"/>	AUCC	<input type="checkbox"/>	GND	<input type="checkbox"/>	MISO	<input type="checkbox"/>	UTG	<input type="checkbox"/>
TXD0 PDI PE1	<input type="checkbox"/>	XCKD AINO PE2	<input type="checkbox"/>	AREF	<input type="checkbox"/>	PFO ADC0	<input type="checkbox"/>	SCK	<input type="checkbox"/>	MOSI	<input type="checkbox"/>
OC3A AIN1 PE3	<input type="checkbox"/>	OC3B INT4 PE4	<input type="checkbox"/>	ADC1 PF1	<input type="checkbox"/>	PF2 ADC2	<input type="checkbox"/>	/RST	<input type="checkbox"/>	GND	<input type="checkbox"/>
OC3C INT5 PE5	<input type="checkbox"/>	T3 INT6 PE6	<input type="checkbox"/>	ADC3 PF3	<input type="checkbox"/>	PF4 ADC4 TCK	<input type="checkbox"/>				
IC3 INT7 PE7	<input type="checkbox"/>	/SS PB0	<input type="checkbox"/>	ADC5 TMS PF5	<input type="checkbox"/>	PF6 ADC6 TDO	<input type="checkbox"/>				
SCK PB1	<input type="checkbox"/>	MOSI PB2	<input type="checkbox"/>	ADC7 TDI PF7	<input type="checkbox"/>	GND	<input type="checkbox"/>				
MISO PB3	<input type="checkbox"/>	OC0 PB4	<input type="checkbox"/>	UCC	<input type="checkbox"/>	PA0 ADO	<input type="checkbox"/>				
OC1A PB5	<input type="checkbox"/>	OC1B PB6	<input type="checkbox"/>	AD1 PA1	<input type="checkbox"/>	PA2 AD2	<input type="checkbox"/>				
OC2 OC1C PB7	<input type="checkbox"/>	NC	<input type="checkbox"/>	AD3 PA3	<input type="checkbox"/>	PA4 AD4	<input type="checkbox"/>				
NC	<input type="checkbox"/>	/RESET	<input type="checkbox"/>	AD5 PA5	<input type="checkbox"/>	PA6 AD6	<input type="checkbox"/>				
UCC	<input type="checkbox"/>	GND	<input type="checkbox"/>	AD7 PA7	<input type="checkbox"/>	PG2 ALE	<input type="checkbox"/>				
NC	<input type="checkbox"/>	NC	<input type="checkbox"/>	A15 CLKD PC7	<input type="checkbox"/>	PC6 A14	<input type="checkbox"/>				
SCL INTO PD0	<input type="checkbox"/>	SDA INT1 PD1	<input type="checkbox"/>	A13 PC5	<input type="checkbox"/>	PC4 A12	<input type="checkbox"/>				
RXD1 INT2 PD2	<input type="checkbox"/>	TXD1 INT3 PD3	<input type="checkbox"/>	A11 PC3	<input type="checkbox"/>	PC2 A10	<input type="checkbox"/>				
IC1 PD4	<input type="checkbox"/>	TXCAN XCK1 PD5	<input type="checkbox"/>	A9 PC1	<input type="checkbox"/>	PC0 A8	<input type="checkbox"/>				
RXCAN T1 PD6	<input type="checkbox"/>	T2 PD7	<input type="checkbox"/>	/RD PG1	<input type="checkbox"/>	PG0 /WR	<input type="checkbox"/>				

J1		J2	
<input type="checkbox"/>	GND	3	1
<input type="checkbox"/>	/RESET	2	
		1-2 => disable	
		2-3 => enable	
		RS232	

Board Layout and Dimensions – Crumb128-CAN is a ~40x30mm² double-sided FR4 PCB (1.6mm). Header holes have 0.9mm diameter, hence most standard round or rectangular pin headers can be mounted. See pictures for details.



Development Tools – Crumb128-CAN is based on the ATmega128 AVR microcontroller, which can be programmed either in assembler (e.g. the original AVR Studio by Atmel: <http://www.atmel.com/avr>) or with several high level languages, including C/C++, Pascal or Basic. There exist several commercial C/C++ compiler suites (e.g. IAR Embedded Workbench or CodeVisionAVR) as well as the WinAVR GNU C/C++ compiler and tools suite (see <http://winavr.sourceforge.net> for details, the bootloader of Crumb128-CAN was developed with WinAVR). A suitable and reasonably priced Basic compiler is BASCOM-AVR by <http://www.mcselec.com>. For a good Pascal environment please go to <http://www.e-lab.de>.

Code Examples – The Crumb128-CAN download page (www.chip45.com/Crumb128-CAN_Downloads) provides free code examples for using the CAN controller.

Further Information – Application notes and data sheets of the onboard components as well as the schematics can be downloaded at <http://www.chip45.com>. The official Atmel AVR homepage is <http://www.atmel.com/avr>. A valuable source of information dedicated to AVR microcontrollers is <http://www.avrfreaks.net>.

Disclaimer – Erik Lins makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein. Erik Lins products are not intended for use in medical, life saving or life sustaining applications. Erik Lins retains the right to make changes to these specifications at any time, without notice. All product names referenced herein are trademarks of their respective companies. chip45.com is a registered trademark of Erik Lins.