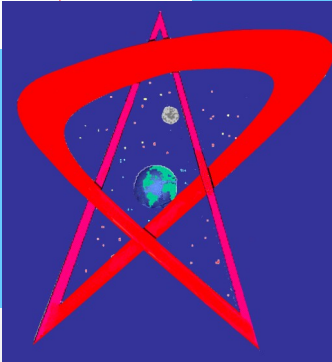


# BROWNS BRAIN

T H E R E ' S N O T H I N G S M A R T E R T H A N A B R A I N !



## PROGRAMMABLE MULTI PURPOSE CONTROLLER BOARD B B 0 1 4



The Browns Brain programmable multi purpose controller board, **BB014**, is designed to serve multiple scenarios. With this product you can easily design a compact and smart device complete with the ability to communicate through industry standard RS232, I2C or use a parallel interface.

Example of embedded application software is included, compatible with: WIN98, NT4, WIN2000 and XP. Here are some of the features:

1. 512 Bytes EEPROM, ATMEGA16 — allows storing modifiable program on board or use as data logger memory
2. Battery backup / charging circuit
3. +7V to 15V power
4. Sample application includes embedded C code
5. PORTS:

- LCD port
- Relay driver port
- debounced switch input port
- LED driver port
- RS232
- I2C
- 2 general purpose 8 bit bidirectional ports

The BB014 has the ability to store and run a program on internal 512 bytes of EEPROM.

```
/*.....*/
/* This uses the initialization of the uc. */
/* Target: ATMEGA16 (Paul W. Brown | October/16/2006 */
/*.....*/
#include "common.h"

void init_devices(void)
{
    unsigned char c;
    CLI();
    //disable all interrupts
    wdr_wdr = 0x0000;
    wdr_counter = 0x0000;
    W_DR = 0xFF;
    DRRM = 0x00; //on reset, set to 0x00
    TPRR = 0x80; //on reset, set to 12.5K = 0x80
    rcount = 0x00; // # of bytes received
    multiread = 0x00; // # of bytes received
    runprogram = 0x00;

    for(i=0x00; i<0x07; i++) Rx_Data[i] = 0x00; //clear the receive array
    for(i=0x00; i<0x07; i++) Tx_Data[i] = 0x00; //clear the rest of the Tx array
    //set up the initial transmit data
    Tx_Data[0x00] = 0x01; //controller board identifier
    Tx_Data[0x01] = 0x02; //code for WDRM (0x00 = 0x00)
    Tx_Data[0x02] = 0x03; //#bytes to be sent
    Tx_Data[0x03] = 0x00; //data1
    Tx_Data[0x04] = 0x00; //data2
    Tx_Data[0x05] = 0x00; //data3
    Tx_Data[0x06] = 0x00; //CRC or CRC16
    Tx_Data[0x07] = 0x01; //identifier

    //RS232 initialization stuff
    //this enables the Rx int & the Tx & Tx are enabled
    //also sets to 8 data bits (always 1 STOP bit and 1 STOP bit)
    UCSRB = 0x08;
    UCSRC = 0x80;
    //this sets the BAUD rate to 19200 b/s
    UBRRH = 0x00;
    UBRL = 0x00;

    //.....

    //Port pins 1 = 0/P, 0 = 1/P
    DDRA = 0x00; //encoder data from motor control PCB, D0-D7
    PORTA = 0xFF; //enable pullup resistors
}
```

with example embedded application,  
with source code. (C code)

[www.brownsbrain.com](http://www.brownsbrain.com)

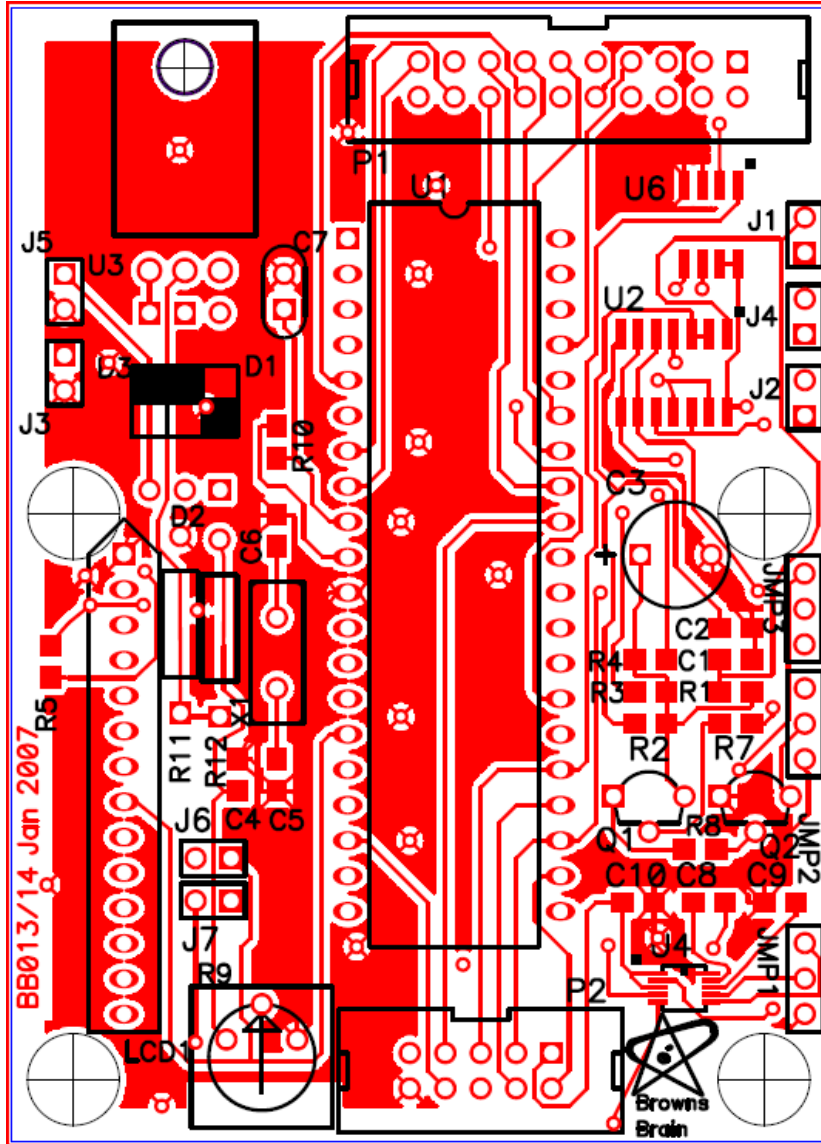
Email: [sales@brownsbrain.com](mailto:sales@brownsbrain.com)

# BROWNS BRAIN

T H E R E ' S N O T H I N G S M A R T E R T H A N A B R A I N !

P1 - Parallel data/control connection

pin19	ENC_RDY'	Not connected	D7	D6	D5	D4	D3	D2	D1	D0	pin1
pin20	+5V	Rd_ENC'	STOP'	FWD/REV/PB4	GN	GN	D1	D1	D9	D8	pin2



J1 - switch input  
 input  
 GND  
 anode  
 cathode  
 anode  
 cathode  
 J4, J2  
 LED connectors

SDA  
 SCL  
 GND  
 +5V  
 Relay on  
 GND  
 JMP3 - I2C connector  
 JMP2 - Relay connector  
 JMP1 - RS232 connection  
 Rx - into this PCB  
 Tx - out of this PCB  
 GND

R9 - LCD Brightness

pin9	Not connected	AREF	PC3 / TMS	PC1 / SDA	+5V
pin10	GND	PD4 / OC1B	Not connected	PC2 / TCK	PC0 / SCL

P2 - general I/O

pin1  
 Pin2