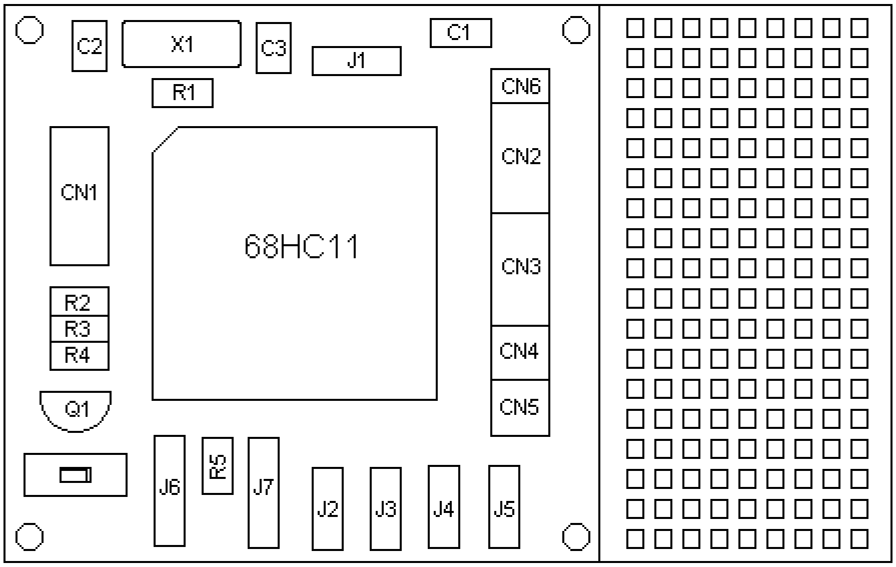
The BOTBoard



Annotated Version by Kevin Ross

Created by Marvin Green
V1.03

INTRODUCTION TO THE BOTBoard.

Thank you for purchasing the BOTBoard. I think you will find the BOTBoard a powerful tool for designing and building small microcontroller projects. The BOTBoard was engineered to help you get your projects up and running quickly, easily and cheaply.

Please take note that this is not a programming guide to the 68HC11, or a tutorial on microcontrollers, or electronics. This manual is an **introduction** to the BOTBoard. You should have some basic knowledge of electronics, microcontrollers and soldering before you tackle this project. Please contact Motorola for technical manuals on the 68HC11 and the free PCBUG development software.

BOTBoard is a 2" X 3", single sided printed circuit board designed around the 68HC11 family of microcontrollers. On the BOTBoard, the 68HC11 is set into single chip mode, and can execute a program from its internal EEPROM or EPROM. All major I/O pins are brought to header's along with a TTL level serial port and SPI port for networking. Four connectors are dedicated to controlling hobby R/C servos, and a prototyping area is also provided.

This board was designed with amateur robotics in mind, with a construction cost approximately. $20. The BOTBoard is adapted from the articles and circuits described in P.A.R.T.S , the Portland Area Robotics newsletter, describing a how to build a single chip robot. The BOTBoards layout is very generic and may be used for general experimentation and project development.

The following pages, contains a list of needed parts, schematics, building instructions, and application notes for the BOTBoard. Please read all instructions before beginning. Enjoy.

Marvin Green 1-25-94

I created the BOTBoard to encourage the building of small robots. It is a tool that I would like to have to help develop robot projects. Special thanks goes to Karl Lunt, Bob Nansel, and Dan Mauch for their great deal of help, and also my wife Coreen for inspiration and enthusiasm.

Special BOTBoard Features:

- Four R/C Servo Ports.(Airtronics XL).
- Reset Switch and low voltage circuitry.
- Auto start jumper.
- 1" X 2" prototyping area,. with power bus strips.
- Networking Port (SPI). With Master/Slave selection.
- Powered RS232 Port connector . for serial communication.
- Pull up resistors on IRQ and XIRQ.
- All chip I/O pins on .100 grid headers.
- Single sided circuit board design.
- Designed for the 68HC11 family.
- Stackable mounting holes.

Features of the 68HC11:

- Low power, less than 20ma.
- Contains 36 I/O pins.
- Built in RAM and EEPROM .
- Eight 8-bit A/D converters.
- RS232 and SPI serial Ports.
- Real time interrupt and COP circuits.
- Able to auto start EEPROM code.
- Timer subsystem with three input captures.
- Five output compares and input accumulator.
- Boot loader program in ROM.

PARTS LIST:

U1	68HC11 or 68HC811 Microcontroller	A, B, C, G	$3 - $25
~~Q1~~	~~MC34064 Low voltage Reset~~	~~A, C~~	~~$2~~
X1	8.00 Mhz Crystal	B, D	$2
R1	1 meg ohm resistor, 1/8 watt	B, C, D	$1
R2, R3, R5	10 K ohm resistor, 1/8 watt	B, C, D -	.30
R4	10 K resistor, 1/8 watt	B, C, D -	.10
C1,C4	.1 uF, Radial type Capacitor	B, C, D -	.15
C1 alternate	10 uF Electrolytic Capacitor	B, C, D -	.50
C2, C3	22 pf Ceramic Capacitor	B, C, D -	.20
SW1	Tiny momentary push button switch. Digi-Key # P8028S	E	.40
P1-P2, P3-P4	Small gauge wire.	E, E	-
PLCC 52 pin socket for U1. Digi Key # A418		B, C, D,	$2
Headers	qty. 1- 26 Pin Header (13x2)	B, C, D, E	$2
Digi Key #	1- 10 Pin Header (2x5)	B, C, D, E	-
929834-10-36-ND..(1x)	5- 3 Pin Header	B, C, D, E	-
929836-09-36-ND (2x)	2- 4 Pin Header	B, C, D, E	-
Jumper	For auto start, jumper on J6	C, D, E	-
Connectors	3M style or equivalent, with .100 spacing.	C, D, E	-
PCBUG	PCBUG11 programming tool, and complete development system, on Motorola BBS.	A	Free

For this project, you need a RS232 converter that converts logic level RS232 signals into true RS232 level signals. See instructions for schematic, and information. Other chips may also be used.

MAX232	RS232 level converter	B, D, F	$2 - $5
	Needs 4 - 10 Mf. caps	B, D, E	
MAX233	RS232 level converter, No caps.	F	$10 - $15

A. Motorola Design Team 800 521-6274
BBS 512 891-3733
- Technical Docs Literature P.O. Box 20912 Santa Clara, CA 94039
- Software PCBUG PCBUG Manual Mail Stop #OE319
6501 William Cannon Dr. W., Austin, Texas 78735

B. B.G MICRO, PO Box 280298, Dallas Texas 75228 214 271-5546

C. Active Electronics, 800 677-8899 for catalog

D. Digi Key, 800 344-4539

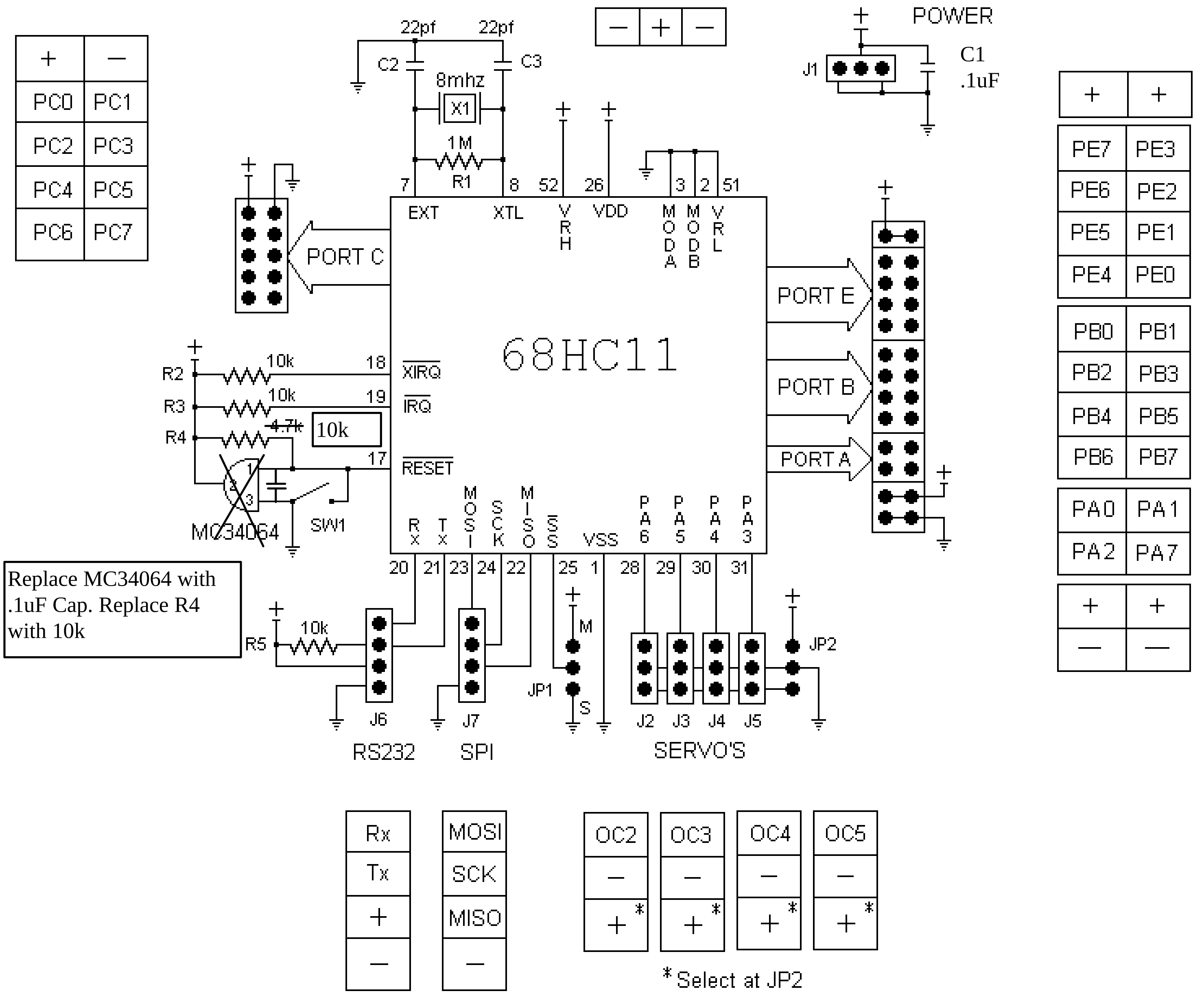
E. Radio Shack, try local area phone book for locations.

F. Maxim I. P., 120 San Gabriel Drive, Sunnyvale, CA 94086 408 737-7600
Maxim 1 800-998-8800.

G: Beall and Glenn Enterprises Rt. 1 Box 242
Lacey’s Spring, AL 35754 800-874-4797

***Shop around and try other vendors for best prices, and supply.**

Schematic of the BOTBoard



~~4.7k~~ 10k

Replace MC34064 with .1uF Cap. Replace R4 with 10k

−	+	−

+	−
PC0	PC1
PC2	PC3
PC4	PC5
PC6	PC7

+	+
PE7	PE3
PE6	PE2
PE5	PE1
PE4	PE0

PB0	PB1
PB2	PB3
PB4	PB5
PB6	PB7

PA0	PA1
PA2	PA7
+	+
—	—

Rx	MOSI
Tx	SCK
+	MISO
−	−

OC2	OC3	OC4	OC5
−	−	−	−
+*	+*	+*	+*

*Select at JP2

Note: *I have had good success replacing the MC34064 with a .1uf capacitor between pins one and three, and changing R4 to a 10k resistor. Please refer to Motorola documentation on reset and low voltage reset.*

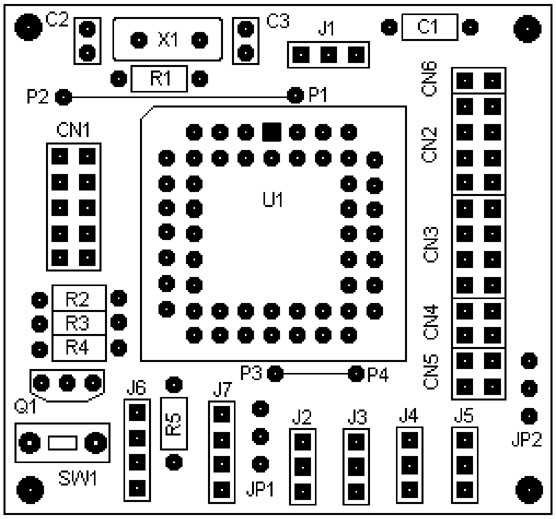
Note: Kevin Ross' kit uses a 10k resistor for R4 instead of 4.7k, and a .1uF cap in place of MC34064

Instructions for Constructing the BOTBoard.

Care should be taken that solder bridges are not created between traces and pads. You should inspect each connection closely. In each step, refer to the circuit board diagram for parts location and placement. You can use masking tape to hold your component secure while you solder. At the end of each step, cut off component leads if needed.

If you prefer, you can remove the prototype area at this point. Its much easier to cut the PC board without its components mounted. This will give you a small 2" X 2" board.

Circuit board shown without prototyping area.



1. The first components mounted to the BOTBoard is the two jumpers, P1-P2 and P3-P4. Simply strip two wires, to the appropriate length and solder into position.

2. Next solder in the resistors starting with R1 (1meg), then follow with R2, R3 and R5 (10k), and last resistor R4 (10k).

3. Mounting the capacitors are next starting with C1 (.1 uf or 10uF. Use 10uF if you plan to run servo motors.), then mount the two capacitors C2 and C3 (22pf).

4. Insert .1uF capacitor C4 in place of Q1. The capacitor should go in pins 1 and 3. Pin 2 (the center pin) will be empty. The MC34064 is not supplied or required.

5. The 52 pin PLCC socket is next. You must get the orientation correct. The *flat inside* corner needs to be toward R1, notice that pin 1 is up.

6. Mount X1 the 8Mhz crystal between C2 and C3.

7. Next solder the three pin power connection header J1. All connector jacks (J*) are single row male headers. The headers have no pin polarization so use caution when connecting to servo's or other devices.

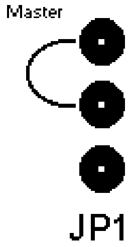
8. Solder the four pin serial connector J6. The bottom two pins provide power for the RS232 converter if needed.

This completes the essential part of construction for the BOTBoard. The BOTBoard is operational at this point. Check all connections and parts placement before continuing.

OPTIONS :

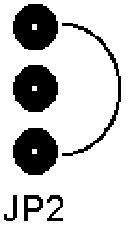
To take full advantage of the Microprocessor I/O, you should mount headers:

CN1 (PORT C),	CN2 (PORT E),
CN3 (PORT B),	CN4 (PORT A is only PA0, PA1, PA2, PA7)



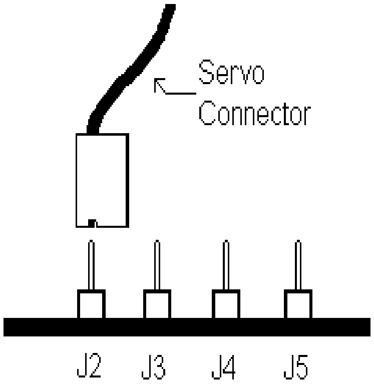
Mount headers to servo connections J2, J3, J4, and J5, then J7 for the SPI networking port.

Note that JP1 may be used to select the master or slave option on the SPI connection. The SS pin can be jumpered to either positive or ground, or the DDRD can be used to set the master/slave option. Refer to your Motorola documentation for details on using the SPI port.



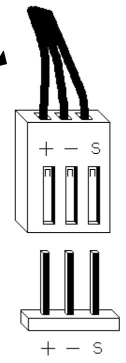
Note from Kevin Ross: Do NOT jumper JP1 (SS). SS is a by directional pin. If you jumper it to positive, and the pin is incorrectly set to an output, you will burn out the pin on the CPU, and perhaps damage other parts of the CPU.

JP2 should be jumpered if you choose to use the BOTBoards **power supply for the servo ports** (as shown). Some servo's, and other devices, may need to have a separate power source. You can use JP2 to connect to another power supply if needed.



IMPORTANT when using SERVOS:

Servo ports J2, J3, J4, J5 are designed to connect directly to an R/C servo. The Royal Titan II or other Airtronics XL compatible R/C servo's are ideal for the BOTBoard. The BOTBoard was designed with an Airtronics compatible connector. The wiring must to be [signal, ground, positive]. If you have a different servo connector, you may need to rewire it. Many servos use the black or green wire as ground, red as positive and the remaining lead as signal.



The Prototyping Area:

The prototyping area has two power bus strips, the top is positive and the bottom strip is negative. The proto typing area can be removed without affecting the BOTBoard.

Simple Test:

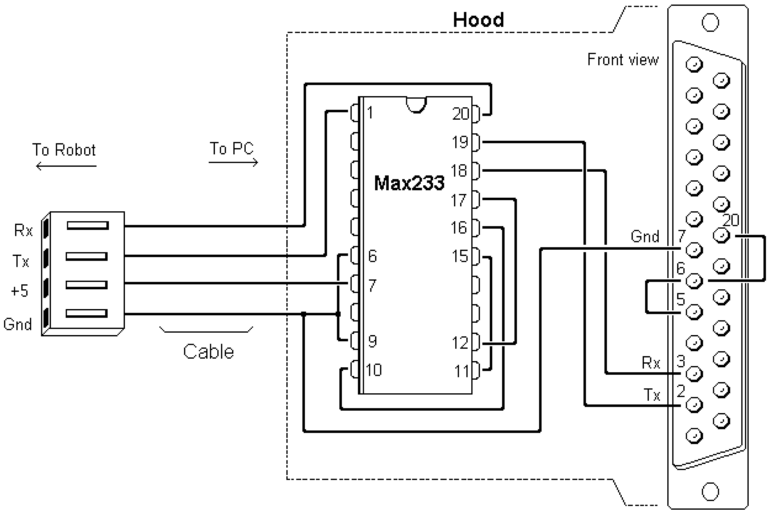
Gently push the CPU in the PLCC socket noting orientation of the chip, then apply power. Use a logic probe to check the crystal connections. If you read a clock pulse, it indicates the CPU is working.

Application Notes:

The BOTBoard is using the 68HC11 in single chip mode. The BOTBoard is designed to perform a task as a node in a larger design, or as a stand alone controller. When using the 68HC11 you have only 512 bytes of EEPROM, this may be limiting. The 68HC811 also works in the BOTBoard, and has 2K EEPROM.

Serial Port:

You will need to connect the BOTBoard to the serial port of a PC to download programs. To do this you must convert the TTL level signals of the 68HC11 to true RS232 levels. Once you build one converter, it may be used to program any number of BOTBoards, just by plugging it into the serial port.



The MAX232 is a 16 pin IC that uses four 1.0Mf capacitors to boost the chips voltage. This chip is very inexpensive and is readily available.

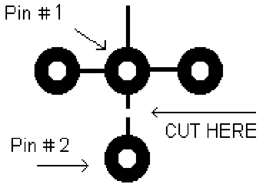
Maxim has recently come out with the **MAX233,** a 20 pin IC that has the advantage of needing **no external capacitors.** One IC does it all. You can fit the **MAX233** in the case of the DB25 connector hood. The female connector is a MTA style plug, or equivalent, that plugs into J6.

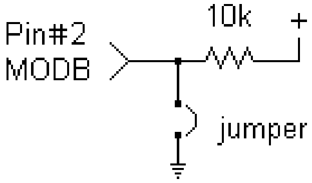
Auto Starting:

To autostart a program in EEPROM on a 68HC11 (A1), you need to connect the transmit and receive lines of the CPU, then do a reset or power up. To do this use a jumper across the **top two pins on J6**. Refer to the Motorola documentation on program execution and keep in mind that the 68HC11 is running in bootstrap mode. *Do not jumper the bottom two power lead pins!*

The BOTBoard is hard wired to operate in **Special Bootstrap Mode**. This mode of operation is fine for most projects, however, you may decide that you need normal **Single Chip Mode**. (Ideal for the 68HC811). Also note that when using **Special Bootstrap Mode**, the interrupt vectors are mapped to RAM. Refer to Motorola documentation.
(*Kevin Ross highly recommends doing this modification. An extra 10k resistor and two pins of the single row header will be available for you to do this on the prototype area). Once installed, downloading to the BotBoard is done by installing the Pin#2 MODB jumper. Removing the jumper allows your program to run.*

The modification for Single Chip Mode requires a jumper, a pull-up resistor, and a cut trace. Follow the Schematic and then set the jumper for Bootstrap Mode, remove it for Single Chip Mode.





Miscellanies:

The reset switch can be mounted to provide a manual reset for the CPU. The BOTBoard can be wire wrapped if you use the appropriate sockets, or it can be placed on a larger prototyping board because all I/O pins are on a .100 grid.

Software:

You can assemble your programs using Motorola's free assembler ASMCH11.EXE, then download the generated *.S19 files to the BOTBoard using PCBUG. PCBUG can also be used to directly program the 68HC11. PCBUG is interactive and has a built in assembler /disassembler with many nice features. You may also use other programming environments, a good source for other options is the Motorola BBS.

Here are some code segments to help you get started. The Motorola documentation has dozens of programming examples. Remember to experiment and have fun.

Example: Toggle bits on the B PORT.

To test PORT B for example, you can run this tiny program to toggle its pins on and off.

```
        clr     $1004       Clears PORT B to $00.
Main    inc     $1004       Adds one to PORT B
        jmp     Main        Creates an infinite loop.
```

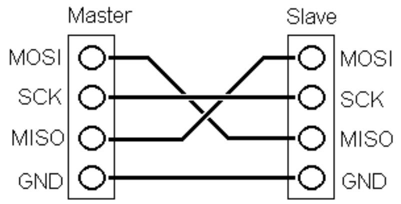
Example: Code to set up Servo on PA6 -J2 and PA5-J3

```
ldd             #$0000      Stop servo's
std             $1018       TOC2 set to 0
std             $101a       TOC3 set to 0
```

```
ldaa            #$60        Bits for OC2 and OC3
staa            $100c       OC1 will control OC2 and OC3.
ldaa            #$00        When OC1 times out it will
staa            $100d       clear OC2 and OC3.
ldaa            #$f0        When OC2 and OC3 time out,
staa            $1020       they will output high.
ldd             #$0000      Get timeout value for OC1
std             $1016       and save in timer reg

ldd    #$f800   Servo value for Clock Wise
std    $1018    set TOC2, PA6 - J2.

ldd    #$f200   Servo value for CCW
std    $1018    Set TOC2, PA6 - J2.
```



Connecting Network Ports

Network Port Wire Diagram:

To connect two BOTBoards through the SPI Networking Port is easy, just connect as shown.

There is so much that you can do by programming the BOTBoard. The BOTBoard was designed to be flexible, powerful and inexpensive. I hope that it gives you a head start on your next project.