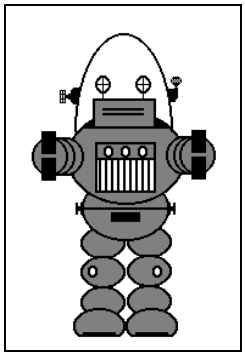
P.A.R.T.S

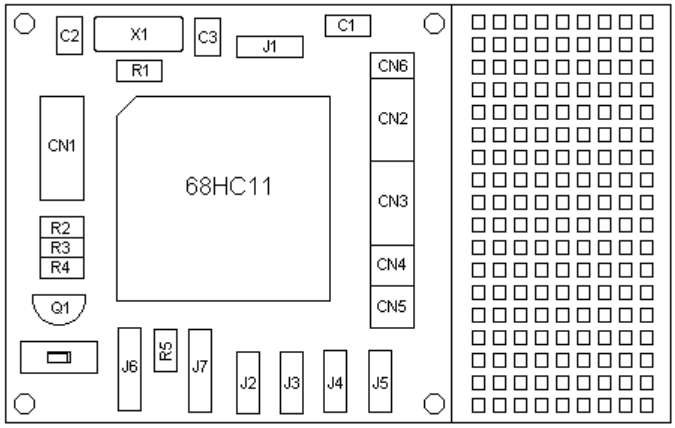


Issue # 09 By. Marvin Green (503) 666-5907.

The BOTBoard is really here! The BOTBoard is a 2" X 3" small robot development board using the 68HC11 in single chip mode. The BOTBoard is designed to be flexible, easy to use, and cheap. It's ideal for a small robot project, or as a node with multiple processors.

Special BOTBoard Features:

. Four R/C Servo Ports.
. Auto start jumper.
. Easy to use Networking Port.
. Powered RS232 Port connector.
. All I/O pins on .10 grid headers.
. Uses the 68HC11 or 68HC811.
. Power supply connector.
. Reset Switch and low voltage circuitry.
. 1" X 2" prototyping area and power bus.
. Master/Slave selection jumper.
. Pull up resistors on IRQ and XIRQ.
. Single sided circuit board design.
. Stackable mounting holes.
. Single chip design.



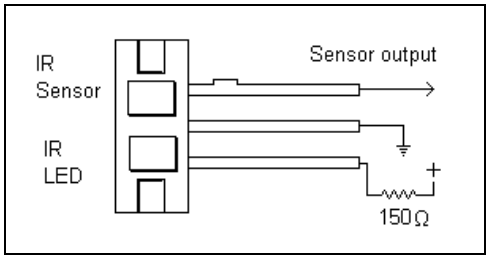
Each board costs only $5.95 in singles or three BOTBoards for $15. Each order comes with instructions, parts list, schematics, and much more.

Contact: Marvin Green 821 SW 14th Troutdale, OR 97060. (503) 666-5907

I found a source for 68HC11's for only $3.00 each. Contact Beal & Glenn Enterprises, Rt. 1 Box 242, Lacey's Spring, AL 35754. The chips have 512B RAM and 512B EEPROM.

Electronic Goldmine at PO. Box 5408 Scottsdale AZ 85261, (602) 451-7454, is a wealth of wonderful electronic parts. Recently I placed an order with them containing many small robot type parts, one of them is a great tiny IR reflective sensor made by Siemans (part# G3355 only .50 each). This sensor contains both a small IR emitter, and a IR detector. It was designed for a bar-code reader, and makes an excellent sensor for a line follower, limit sensor, or as a tachometer on robot wheels.

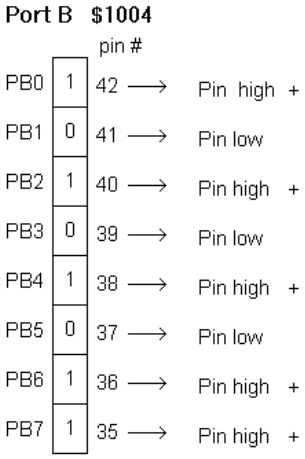
Even though this part is very small, only 1/4 inch wide, it displayed some good characteristics. The sensor reacted well to a black line at 1/2 inch away, and with only three pins is very easy to use.



Using the 68HC11

In the next few issues, I will cover some very basic features of using the 68HC11 microcontroller, starting with some basic input output functions. Most of the functions described pertain to the 68HC11 in single chip mode, but the ideas covered also pertain to other microcontrollers. The examples will be givens in assembly level code. These examples can be programmed with PCBUG11.

The 68HC11 has a total of 52 pins, 38 of those pins can be used in some sort of input/output function. It is possible with computer code to control an individual output pin (a bit) to be either a positive voltage or ground. You can also read a single pin as input to see if that pin is positive or negative. Input, can be as simple as a switch or more complex like a light sensor.



Each I/O pin is represented by one bit. A group of eight I/O pins represents one byte. This byte of I/O it is called a register, each register has an address. The address is the location of the register. Think of the register address like a street address.

For example take port B on the 68HC11. This port is an output port, and the pin numbers are 35 - 42. The address for this port is at $1004 ($=Hexidecimal, %=Binary). If I store a $FF, (%11111111) into the register at address $1004, the B port, all the output pins go high. If I store a $00, (%00000000) into $1004, all the output pins go Low. If I want to turn on just one pin, pin number 41, then I would store $02, (%00000010) at $1004. The assembler code segments would look like:

```
LDAA #$00          *  Load Accumulator with Zero.
STAA $1004         *  Store the value from the accumulator into the B port.
                      Sets all pins to zero.

LDAA #$FF          *  Load Accumulator with 255.
STAA $1004         *  Store the value from the accumulator
                      into the B port..  Sets all pins high.
```

Next is an input port, port E. If you read the contents of the register at $100A (pins 43 - 50), you will read the current state of the input pins. If a pin is high you will read a 1, if it's ground you will read a 0. For example, if all the pins are high you will read a $FF, (%11111111), and if all the pin are high except pin 50, which is low, you will read a $7F, (% 01111111). The code segment might look like this:

```
LDAA $100A   *  Load Accumulator the value from
the
STAA $1004   *  register at address $100A.
             *  Store the accumulator at port B.
```

The 68HC11 technical manual will help you with all the in's and out of working with input output ports.