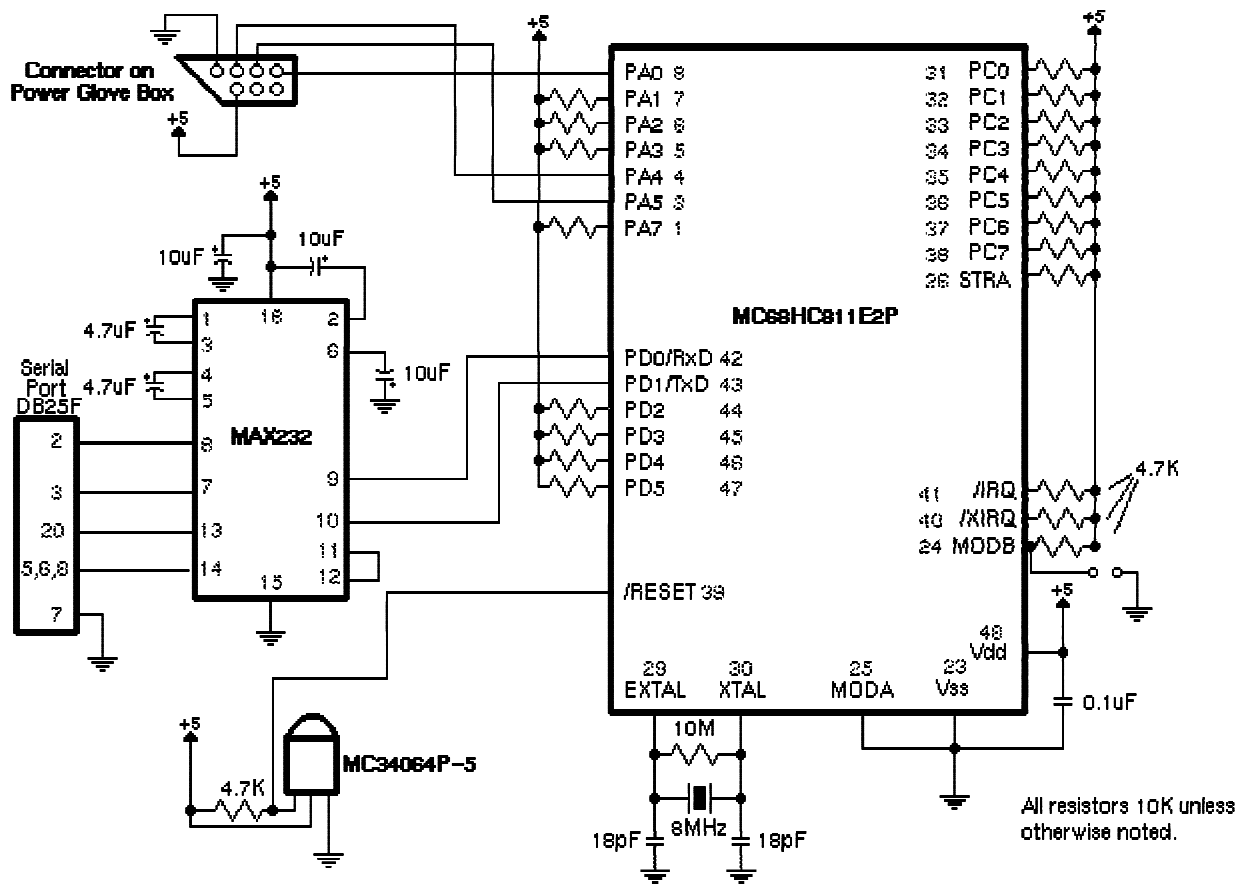


68HC11 Based Power Glove Interface - Ron Menelli 11/19/91



*** Notes on the 68HC11 Power Glove project - 1/14/91

The project consists of the following files:

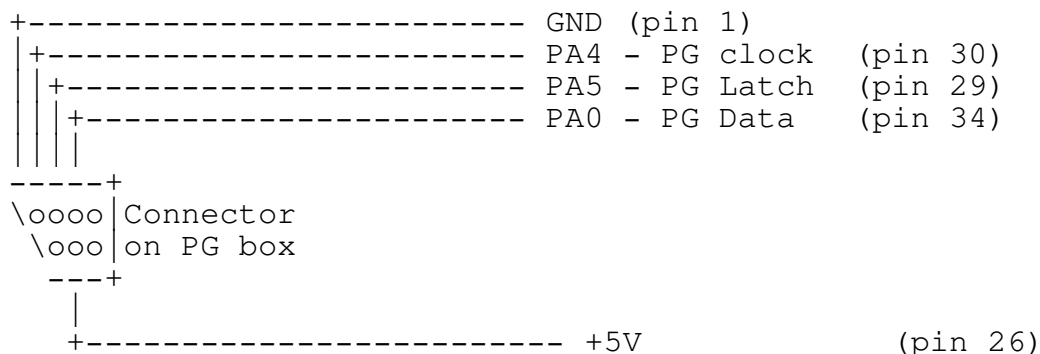
pg.asm	The actual 68HC11 source code for both versions
pg.obj	The 68HC11 object code in DAsm format (This is set up for the standalone version)
pg.s19	The 68HC11 object code in S19 format (This is set up for the EVB version)
eeload.asm	Program used to load an HC11's EEPROM
eeload.o	Raw binary of above program
pg.gif	GIF schematic of the standalone circuit
amiga/pgread.c	Amiga Lattice C source code for the test pgm
amiga/pgread	Executable for above program
amiga/dload.c	Amiga Lattice C source code for bootloader pgm
amiga/dload	Executable for above program
msdos/pgread.c	MS-DOS Turbo C source code for the test program
msdos/pgread.exe	Executable for above program
msdos/dload.c	MS-DOS Turbo C source code for bootloader pgm
msdos/dload.exe	Executable for above program
msdos/egavga.bgi	File required by pgread for EGA or VGA display
msdos/cga.bgi	File required by pgread for CGA display

This code is designed for both the 68HC11EVB board from Motorola, and a standalone board that I have enclosed the schematics for.

How to get it working

* Using the 68HC11EVB

- Make a cable from the power glove to the EVB. The pin numbers on the EVB side refer to the 60 pin expansion connector on the EVB. (I'm not sure these are the same pin numbers on the EVB-U's connector, however.)



- Connect the EVB terminal port to a serial port on your PC (or whatever you have).
- With everything now connected, download the file pg.s19 into the onboard RAM following the directions in the EVB manual.
- From the BUFFALO monitor prompt, type "g d800" to start the program. The glove should beep one or two times, and then the lights on the sensor array should begin to flicker.
- If you only have one of the serial ports on the EVB connected to your machine, switch the cable to the other (host) port on the EVB. If you have both connected, make sure you're monitoring the other serial port now.
- The EVB is now sending data from its host port (not the terminal port!) to your serial port at 9600 baud. If you have an IBM compatible or an Amiga, you can use the "pgread" command to test out the interface. See below for details.
- Since the program is being stored in RAM, you must go through the loading procedure every time you turn on the EVB.

* Using the standalone board

- Build the standalone board described in the schematic file "pg.gif". Please note that the pin numbers on the CPU refer to the 48 pin DIP version of the 68HC811E2, not the PLCC one.
- Make sure the files dload (whichever is appropriate for your machine), eeload.o, and pg.obj are in your current directory.
- Close the jumper on the standalone board to activate bootstrap mode. Connect the serial port on the board to a serial port on your machine. Power up the board.
- Run the command "dload -c pg.obj". The MS-DOS version defaults to COM1

(change it by using the "-c <com_port_#>" option), and the Amiga version defaults to serial.device, unit 0 (change it by using the "-d <device_name>" and the "-u <unit_number>" options). This command sends the program code to the EEPROM in the 68HC811 chip.

- Power down the board, remove the jumper, connect the power glove to it, and power up the board. The glove should beep one or two times, and then the lights on the sensor array should begin to flicker.
- The board is now sending data from its serial port to your serial port at 9600 baud. If you have an IBM compatible or an Amiga, you can use the "pgrd" command to test out the interface. See below for details.
- Since the program is stored in EEPROM, this only needs to be done once. Every time you turn it on (after the first), the board will begin executing the Power Glove interface code. Make sure the jumper is no longer shorted!

Notes on pg.asm

The Power Glove interface code is designed to take the dirty work out of talking to the Power Glove. The host computer sends single character commands, and the interface board sends back the raw data from the Power Glove. The format is almost exactly the same as the actual glove data itself:

0xA0 (flag only sent in continuous data mode)
X (single byte position -127..127)
Y "
Z "
rot (single byte 0..11 in 30 degree increments)
fingers (packed byte containing two bits for each of four fingers)
keys (single byte of keypress data)

In continuous mode (the default), the seven byte packet is sent over and over again, with the 0xA0 flag indicating the start of the data. In request mode, the board only sends data when prompted to, and then only sends the last six bytes of the packet. Also, there is a user selectable deglitching mode that filters out some of the noise in the X and Y directions (thanks to Dave Stampe for that code!). Only the hysteresis deglitching routine is used, due to the fact that I couldn't get the other routine to work correctly! The commands used to talk to the board are the following:

C Start continuous data mode (default)
R Start request data mode
? Request a data packet when in request mode
+ Turn on hysteresis deglitching mode (default)
- Turn off hysteresis deglitching mode

Note: At the present time, the serial port interface is fixed at 9600 baud.

The Power Glove interface code was written for the DAsm assembler on the Amiga. The Motorola Freeware Assembler doesn't have macros, so it won't assemble on it without some work. If you modify this code, please let me know so I can see if I would like to add it to a future version.

When assembling the code, make sure that the constant 'EVB' is set to 1 if you are assembling the EVB version, or it is set to 0 if you are doing the standalone version. This insures that the correct starting address is used, and

that the reset vectors are present for the standalone version.

Notes on dload and eeload.o

These two programs are used to get the Power Glove reading code into the standalone board. They are not needed for the EVB version.

By powering the standalone board up with the jumper shorted, the HC11 processor enters the special bootstrap mode. In this mode, it waits for a character to come over the serial port, sets its baud rate based on the character, and then places the next 256 characters received on the serial port into the 256 RAM locations. Next, a jump to location 0 in RAM is executed.

The program "dload" sends a 256 byte or less program to an HC11 in bootstrap mode in the proper manner. This file is expected to be in raw binary format. There is an EEPROM mode that first sends the program eeload.o to the HC11. The HC11 then runs this program, which reads an object file in, and programs it into the internal EEPROM (\$F800 - \$FFFF in the 68HC811E2 processor). The file format that EEPROM mode expects is the following:

```
Byte 1: LSB of start address
      Byte 2: MSB of start address
      Byte 3: LSB of length
Byte 4: MSB of length
      ... (length # of bytes follow)
Byte ?: LSB of start address
Byte ?+1: MSB of start address
Byte ?+2: LSB of length
Byte ?+3: MSB of length
      ... (length # of bytes follow)
.
.
.
```

This format is output by the public domain DAsm assembler for the Amiga. Other assemblers will probably need a conversion utility. Please note that the pre-assembled file pg.obj is in this format already.

The syntax for the dload command is the following:

(MS-DOS syntax)

```
dload [-e] [-c <com_port#>] <filename>
```

- e indicates EEPROM mode
- c indicates which COM port to use. Defaults to 1 for COM1:
- filename is the raw binary file to download to RAM, unless
 - e is specified, in which case it indicates the DAsm format binary file to download to EEPROM

(Amiga syntax)

```
dload [-e] [-d <device_name>] [-u <unit_number>] <filename>
```

- e indicates EEPROM mode
- d indicates which device to use. Defaults to serial.device
- u indicates which unit number to use. Defaults to 0.
- filename is the raw binary file to download to RAM, unless
 - e is specified, in which case it indicates the DAsm format binary file to download to EEPROM

Notes on pgrad

pgrad is a very simple program to test your power glove setup. The glove data is displayed in the upper left corner, and a graphic display shows the X, Y and Z positions of the glove. The following buttons on the glove work during the program:

Center	Centers the glove
Start	Stops the program (ironic, eh?)
A	Turns on hysteresis deglitching mode (default)
B	Turns off hysteresis deglitching mode

=====

I'm open to suggestions for any further features - let me know what you think!

Ron Menelli
menelli@tellabs.com