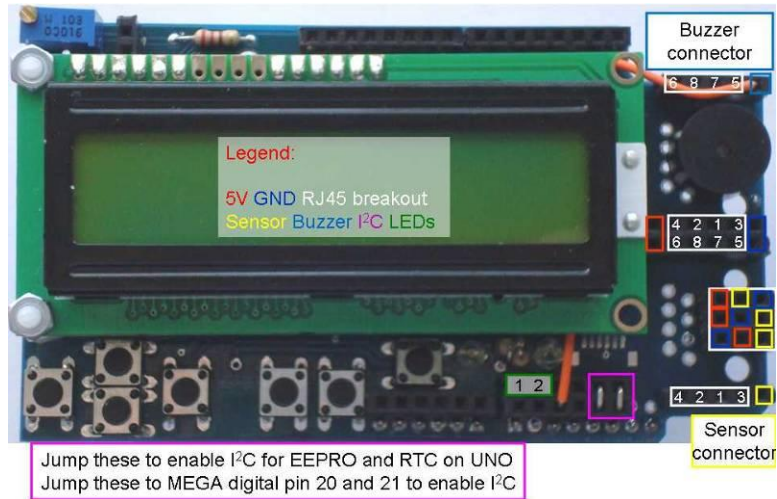


# Phi -2 shield Documentation



*Last reviewed on 5/17/2012  
John Liu*

## Table of content

1.	<b>Introduction:</b>	2
2.	<b>List of functions:</b>	2
3.	<b>Complete projects with code to get you started:</b>	3
4.	<b>Possible projects to get you thinking:</b>	3
5.	<b>Changes from Phi-1 shield and new features:</b>	3
6.	<b>Parts list:</b>	5
7.	<b>Arduino pin usage on the shield:</b>	6
8.	<b>List of accessories:</b>	7
9.	<b>Before assembling:</b>	8
10.	<b>Assemble the shield:</b>	8
11.	<b>Testing the shield:</b>	13
12.	<b>Software:</b>	14
13.	<b>Application notes:</b>	16
14.	<b>Pictures:</b>	17
15.	<b>FAQ:</b>	19

## 1. Introduction:

*Congratulations on getting the best all-purpose interactive shield for Arduino! You are not just getting a piece of hardware; you've just got the best interactive user interface software package, phi\_prompt, and the best support!*

The Phi-2 shield fits Arduino UNO, Duemilanove, MEGA1280, or MEGA2560. It also fits most Arduino clones that are pin-compatible with the above original boards. You can also use it with the official Arduino Ethernet shield with SD card. The Phi-2 shield is revised from the popular Phi-1 shield, which has gained its popularity among Arduino users and instructables.com.

I have developed many complete projects with fully-functional codes, interactive LCD menus, available for download at this location: <http://liudr.wordpress.com/phi-2-shield/>. Several projects such as the Morse code encoder/decoder and the alarm clock don't require anything other than an Arduino board and the Phi-2 shield so you can load up the code and you're ready to enjoy!

The phi\_prompt interactive user interface software library is designed for phi shields. With this library, you can construct menus of various styles, ask user to select items from lists (such as pasta), input numbers (such as RMPs) and texts (such as file names). Find out more about the library at: [http://liudr.wordpress.com/libraries/phi\\_prompt/](http://liudr.wordpress.com/libraries/phi_prompt/)

The ground work of interactive LCD menus with buttons has already been laid for you. You will get a head start with sample codes or simply take parts of the code, such as buttons class, menus, input panel, GPS, sonic ranger, etc to quickly construct your own project.

The Phi-2 shield is designed for uses from prototyping to permanent installation with your finished projects. You can purchase a Phi-duino, an Arduino compatible board, to minimize your cost without any need to redesign your work.

The possibility is endless.

## 2. List of functions:

- ✓ Compatible with Duemilanove, UNO, MEGA, and Ethernet shield with SD card
- ✓ 16X2 LCD or 20X4 LCD character display
- ✓ 6 push buttons arranged in arrow keys and two more on the side
- ✓ 2 RJ45 ports for up to 16 long and robust connections to sensors or devices
- ✓ Buzzer to play tones, ready for any Arduino pin, in place of RJ45 port 1
- ✓ Sensor block for many common sensors, in place of RJ45 port 2, temperature, pressure, light, magnetic field etc.
- ✓ Two 3mm LED indicators, ready for any Arduino pins

- ✓ Real time clock with battery backup keeps the time when Arduino is turned off
- ✓ Optional EEPROM ([24LC256](#) /[24LC512](#)) to log and keep data when power is off
- ✓ GPS connector and breakout for [EM-406A GPS module](#) sold at sparkfun.com
- ✓ Reset button for Arduino
- ✓ All Arduino pins, brought out for maximal flexibility
- ✓ The I2C bus headers next to analog pins 4 and 5 for both UNO and MEGA

### 3. Complete projects with code to get you started:

- ☑ Alarm clock
- ☑ Testing all functions (gives you a sample of how to use everything)
- ☑ GPS interactive data logger
- ☑ Mores code trainer
- ☑ Car reverse obstacle sensor
- ☑ The list keeps growing...

### 4. Possible projects to get you thinking:

- ☆ Standalone or PC data logger
- ☆ Lab data acquisition system (Physics, Chemistry etc)
- ☆ Weather station
- ☆ Input or operating panel, like security panels or garage door opener
- ☆ Handheld GPS
- ☆ Tweet update display (with Arduino Ethernet shield)
- ☆ Interconnected network of arduinos over serial ports
- ☆ The list goes on...

### 5. Changes from Phi-1 shield and new features:

Old: Two RJ11 jacks. Power, GND and one analog channel were hardwired to each jack.

New: Two RJ45 jacks replace the two RJ11 jacks. All 8 pins on each jack are broken out and there are 5v and gnd near each jack for easy access to power and ground. It's more work on the user but offers great flexibility.

-----

Old: Buzzer was hardwired to analog pin 2. An LED was hardwired to analog pin 3.

New: The single LED is removed and replaced by two smaller ones above analog pin headers. The LEDs, the buzzer and sensor block are connected to female headers, so you can jump them to any pin. It's more work on the user but offers more flexibility.

-----

Old: The real time clock and EEPROM chips were hanging below the board.

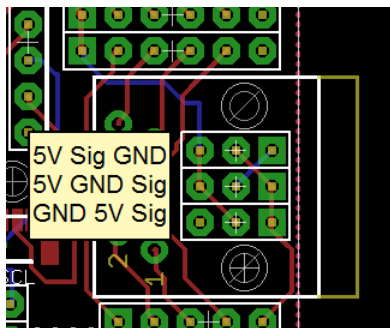
New: The two chips are now on top of the board to give more clearance below the board to work better with more shield boards.

-----  
 Old: The SDA and SCL of the I<sup>2</sup>C bus were hardwired to Arduino analog 4 and 5.

New: The SDA and SCL lines are brought to female headers, next to the analog 4 and 5. Jumping them to analog pins 4 and 5 works with Duemilanove and UNO, while jumping them to 20 and 21 works with MEGA. You can still use analog pins 4 and 5 on MEGA.

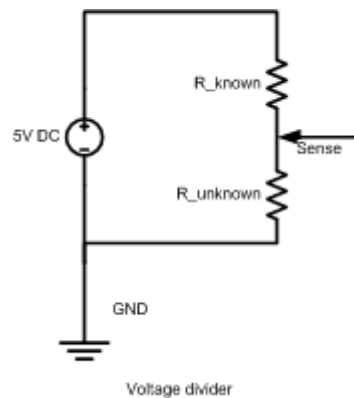
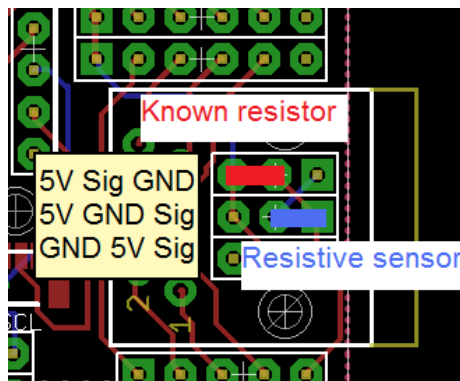
-----  
 Old: One 5mm LED was hardwired to Arduino analog 3.

New: A sensor block replaces it. The sensor block adapts to many analog and digital sensors. The three rows of 3-pin female headers are connected in the following way:



Many sensors have 3 pins, 5V, GND, and signal, such as temperature sensor LM35 or LM36, Hall Effect sensors. With this sensor block, you can tinker with any of them without having to set up a breadboard. To sense analog IC, hook the bottom right pin on the 5-pin header to an analog pin. You can also use the sensor block for purely resistive 2-pin sensors, such as thermistors for temperature, resistive pressure sensors, light-dependent resistor for light, Hall Effect magnetic sensors etc. Just hook the sensor in series with these sensors in the following diagram and use the formula below:

$$R_u = \frac{V}{V_o - V} R_k = \frac{\text{analog reading}}{1024 - \text{analog reading}} R_k$$



## 6. Parts list:

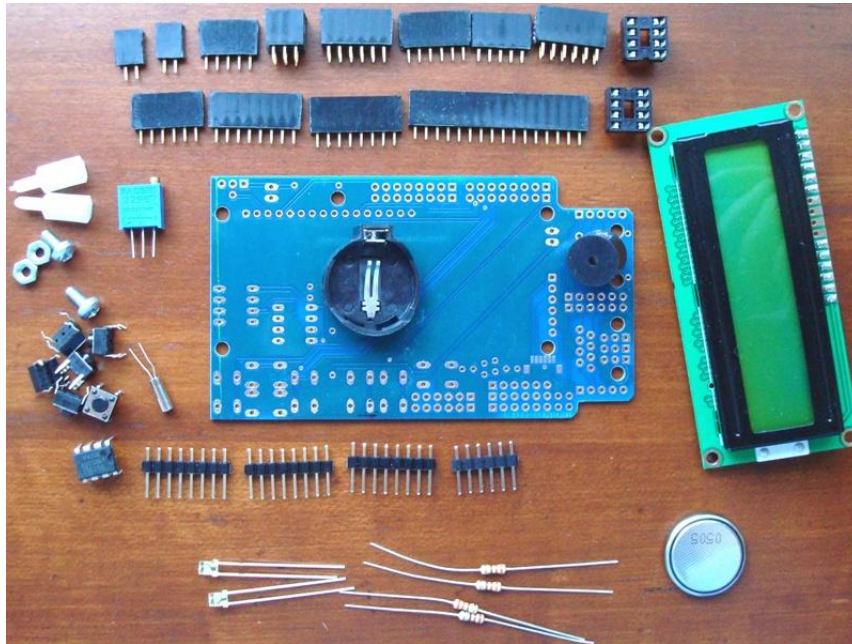


Fig. 6-1 Unassembled 16X2 shield with all parts (standoffs are optional).

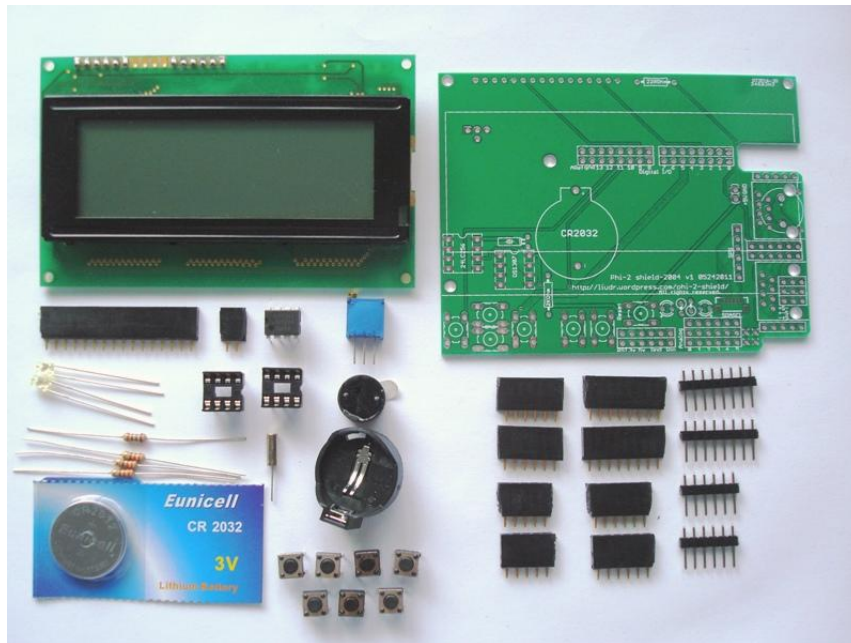


Fig. 6-2 Unassembled 20X4 shield with all parts (user will need to cut female headers to size, later version features stacking headers with long pins).

Link	Item	Qty
<a href="http://www.dipmicro.com/store/LCD-1602A-Y">http://www.dipmicro.com/store/LCD-1602A-Y</a>	Character LCD	1
<a href="http://www.dipmicro.com/store/DS1307N">http://www.dipmicro.com/store/DS1307N</a>	Real time clock	1
<a href="http://www.dipmicro.com/store/XC4-32768">http://www.dipmicro.com/store/XC4-32768</a>	32.768KHz crystal	1
<a href="http://www.dipmicro.com/store/HDR40X1F">http://www.dipmicro.com/store/HDR40X1F</a>	Female pin headers	3
<a href="http://www.dipmicro.com/store/HEADS40T">http://www.dipmicro.com/store/HEADS40T</a>	Male pin headers	1
<a href="http://www.dipmicro.com/store/ICS308">http://www.dipmicro.com/store/ICS308</a>	8-pin DIP sockets	2
<a href="http://www.dipmicro.com/store/BH2025">http://www.dipmicro.com/store/BH2025</a>	CR2032 battery holder	1
<a href="http://www.dipmicro.com/store/MJTP1230">http://www.dipmicro.com/store/MJTP1230</a>	Buttons	7
<a href="http://www.dipmicro.com/store/R4V10-3296">http://www.dipmicro.com/store/R4V10-3296</a>	10KOhm Trimpot	1
<a href="http://www.dipmicro.com/store/BAT-CR2032">http://www.dipmicro.com/store/BAT-CR2032</a>	CR2032 coin battery	1
<a href="http://www.dipmicro.com/store/LED3RD">http://www.dipmicro.com/store/LED3RD</a>	LED	2
<a href="http://dipmicro.com/store/DBX-01PN">http://dipmicro.com/store/DBX-01PN</a>	Speaker	1
<a href="http://dipmicro.com/store/R2F15-4">http://dipmicro.com/store/R2F15-4</a>	150Ohm resistor	1
<a href="http://dipmicro.com/store/R2F33-4">http://dipmicro.com/store/R2F33-4</a>	330Ohm resistor	3

Table 1. Parts list.

## 7. Arduino pin usage on the shield:

As a user, you have three levels of pin access:

- 1 – Free access, the pins are not hooked to anything and you can use them for anything
- 2 – Shared access, some hardware on the shield is connected to these pins so you either choose to use these pins to access the connected hardware or disconnect the hardware and use the pins for general purpose
- 3 – No access, some hardware on the shield or Arduino is connected to these pins and can't be disconnected so you can't use the pins for general purpose

If you plan not to use the SPI bus for Arduino Ethernet shield you can use the pins 12 and 13 for general purpose.

If you plan to use less buttons, you can leave the buttons on the shield, just don't press them so their pins (4, 5, 10, 11, 14, 15) can do other stuff.

If you plan not to use EEPROM and the DS1307 RTC, disconnect jumpers between SCL/SDA analog pins 4 and 5.

If you don't use the LCD, all its pins (2, 3, 6, 7, 8, 9) can be used for general purpose.

You can't use Arduino pins 0 or 1 as they are serial pins that will affect program upload.

Free access	digital 16, 17
-------------	----------------

Shared access

Hardware	Arduino pin
I2C bus (RTC, EEPROM)	18, 19
SPI bus (Ethernet shield)	12, 13
Button Up	5
Button Down / Ethernet CS	10
Button Left	11

Button Right / SD card CS	4
Button B	14
Button A	15
LCD RS	8
LCD EN	9
LCD D4	7
LCD D5	6
LCD D6	2
LCD D7	3

No access	
TTL Serial RX, TX	Digital 0,1

Table 2. Pin usage.

## 8. List of accessories:

The compatible EEPROMs are 24LC256, 24LC512, or 24LC1024.

The compatible GPS module is the EM-406A, sold at [sparkfun.com](http://sparkfun.com). You need a connector from them as well:

GPS connector: <http://www.sparkfun.com/products/579>

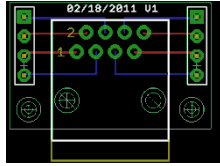
EEPROM: <http://www.sparkfun.com/products/525>

GPS module: <http://www.sparkfun.com/products/465>

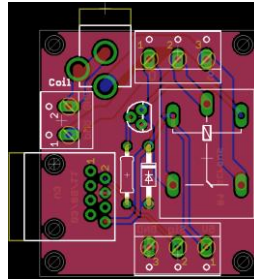
To make use of the RJ45 connections, you can choose the following boards. You may download the design files of the following boards from my Phi\_prompt Google code page and make them at a PCB fab house..

Board	Function
RJ45 breakout board	Breaks out all 8 pins on an RJ45 jack in standard 0.1" pins. You can solder it to your main circuit board or plug on you breadboard to pass connections from Phi-2 shield via Cat-5 wires.
Relay board with RJ45 connector	Provides space for a relay and screw terminals to control high power stuff This board also has a power barrel to easily power an electromagnet or else
RJ45 splitter board	Splits one RJ45 connection to up to six branches to control more devices or sense more sensors.

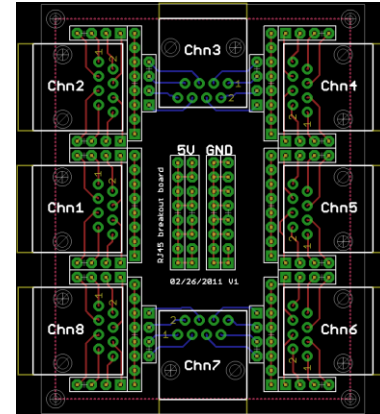
Table 3. Accessories.



RJ45 breakout board



RJ45 relay board



RJ45 splitter board

## 9. Before assembling:

You will need a basic soldering iron with a sharp tip to assemble the shield like this on from RadioShack:

<http://www.radioshack.com/product/index.jsp?productId=2062758>

Purchase some 0.032" diameter solder as well as the included solder runs out fast.

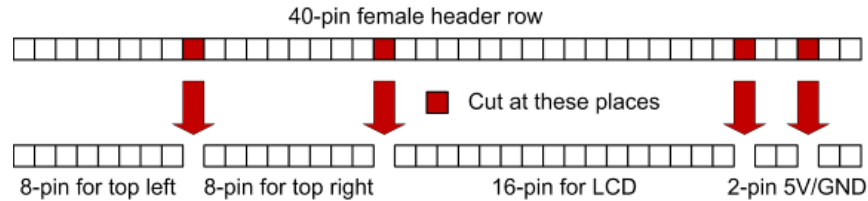
A buzzer and a sensor block or two RJ45 ports or a mix? The buzzer and RJ45-port1 share space. The sensor block and RJ45-port2 share space. You could use the buzzer as an alarm or maybe play a few mono-tunes (not MP3 quality). You could use the sensor block to test out many sensors. See section 3 under changes and new features. On the other hand, the RJ45 ports give you two robust connections to external sensors and controls. You can always de-solder with solder wick or de-soldering iron.

## 10. Assemble the shield:

Before you solder, make sure where the male pins go and where the female pins go and which side of the board is front. All the assembling pictures are in the end of this section. Please follow very closely! If you receive the later 20X4 LCD version with separate female pins, you only need to break away less pieces as you need. The steps to assemble 20X4 version is very similar. There is a picture of the completed 20X4 shield in the end.

1. Break away one 16-pin, two 8-pin, two 2-pin female headers from a row of 40 as shown below:





2. With a second row of 40, break away five 6-pin female headers and one 5-pin header
3. With a 3<sup>rd</sup> row of 40, break away one 5-pin one 6-pin and three 3-pin female headers.
4. Break the male headers into two 8-pin rows and four 6-pin rows.
5. Identify the front side with Fig. 10-1 (button pads on bottom left).
6. If you received a later version of 20X4 LCD shield, you may skip 6 and 7 and solder female stacking headers with long pins in step 9. Male headers are the lower rows among the rows of holes. They need to be soldered below the board since they will plug into the Arduino female headers. Secure two top rows of 8-pin headers with clip or tape (red rectangles in fig. 10-1). Solder the first and the last pin to secure it. Remove clip or tape and solder the rest of the pins.
7. Male pins on the bottom 6-pin rows. The two rightmost pins are for MEGAs only.
8. (Optional Fig. 10-4 red) Carefully tape the GPS connector, leaving space to solder its side. Once soldered on, remove tape and use extreme care while soldering in order not to short the connections or melt plastic.
9. Female headers need to be soldered above the board. Use tape to secure the female headers in place. To secure the headers, wrap tape around it, even if it means covering most of its pins. Solder one pin to the board to secure the row and then you can remove the tape to access the rest of the pins. In Fig. 10-4, secure two 8-pin headers on top, two 6-pin rows on bottom, two in middle right, one 5-pin row on top right. In bottom right, I secured two rows together. It's a lot easier to just do one row at a time.
10. Fig. 10-5. Female pins on top left, top right, both 2-row. Sensor block 3-pin headers (solder one row at a time). See finished result in Fig. 10-6
11. Fig. 10-7, top left LCD, bottom right 5-pin and 6-pin GPS (optional, red) headers.
12. Fig. 10-8, Push all buttons in. Tape potentiometer (red), Dip sockets and crystal (orange, crystal above right socket bending leftwards), LEDs and two 330 Ohm resistors (yellow, two LED '+' long legs on far apart), buzzer (green, + pin on top), LCD resistor (purple 150 Ohm). Battery holder (white) is held by double sticky tape.
13. Solder male headers on LCD pins 1-6, and 11-16. You don't need pins 7-10.
14. Jump the I2C pins as indicated in Fig. 14-8 for UNO or Mega
15. RJ45 jacks. Make sure you push the jacks all the way in before soldering.
16. Insert battery with "+" pointing up. Insert DS1307 with half circle notch pointing up in right socket; do same for 24LC256 or 24LC512 EEPROM for left socket.

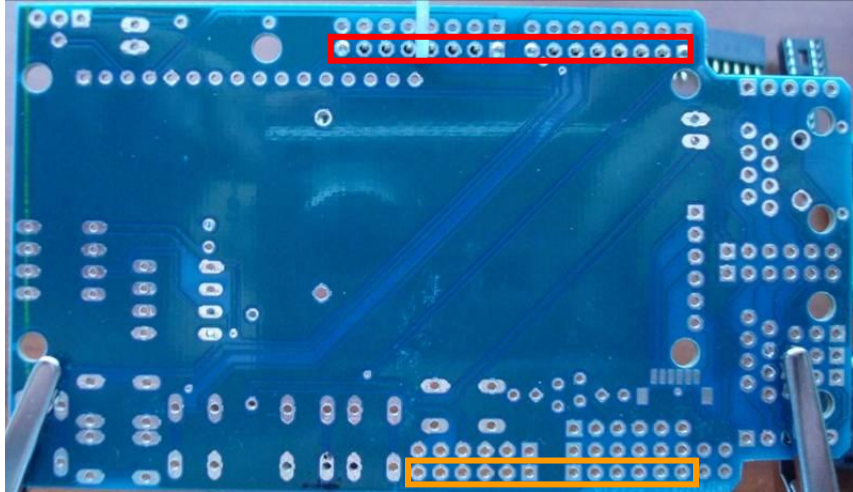


Fig. 10-1 Top side of PCB. Use a clip (top) to secure male pins.



(a)



(b)

Fig. 10-2 Solder the far pins to secure the row. Then solder other pins. (a) Back side of two male pins (red). (b) Back side of bottom male pins (orange).

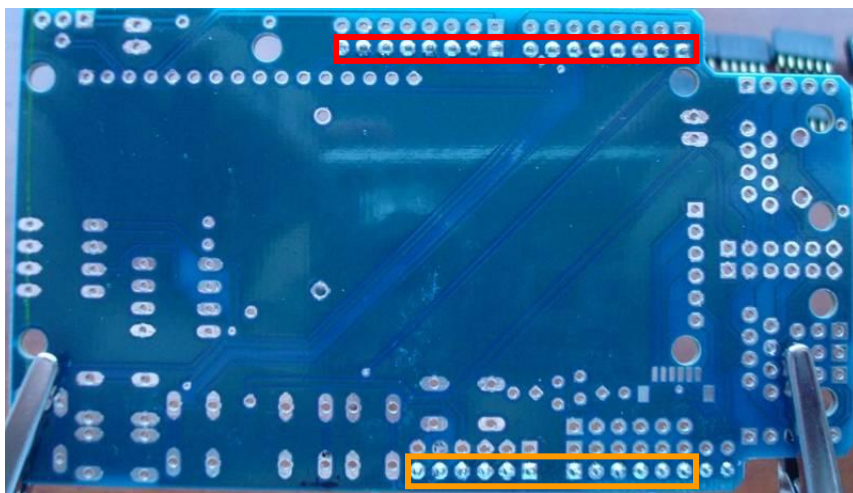


Fig. 10-3 All male pins are soldered.

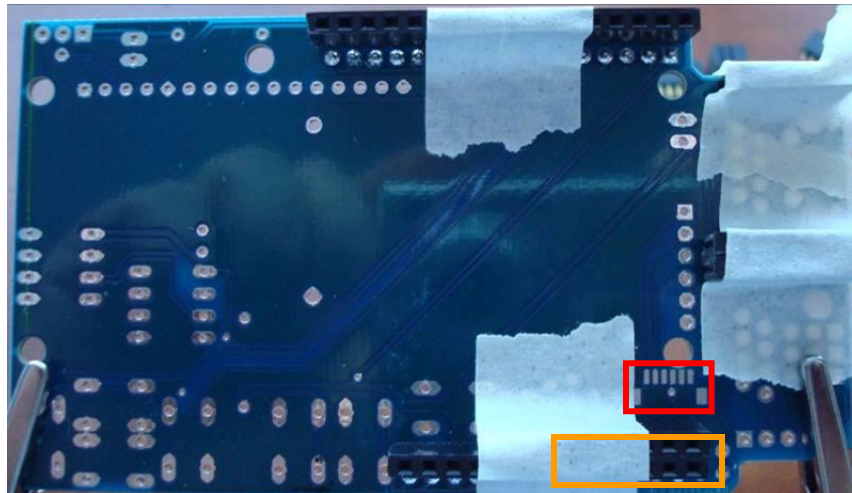


Fig. 10-4 Use masking tape to secure female pins before soldering.

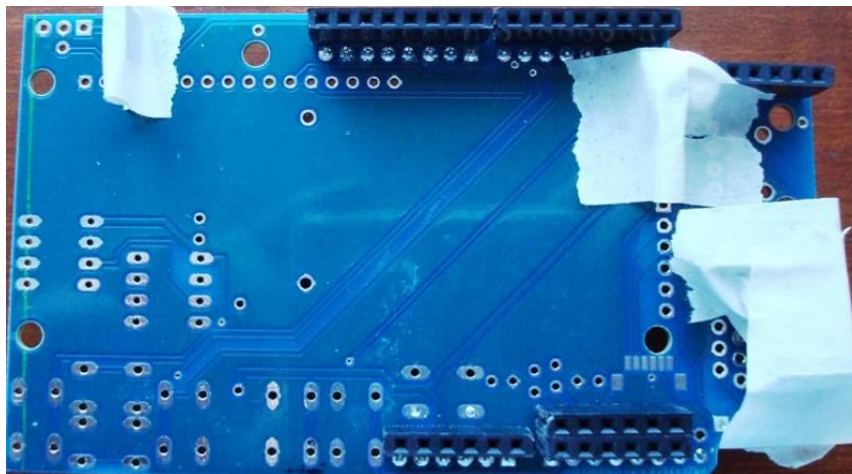


Fig. 10-5 Secure more female pins with tape after Fig. 10-4.

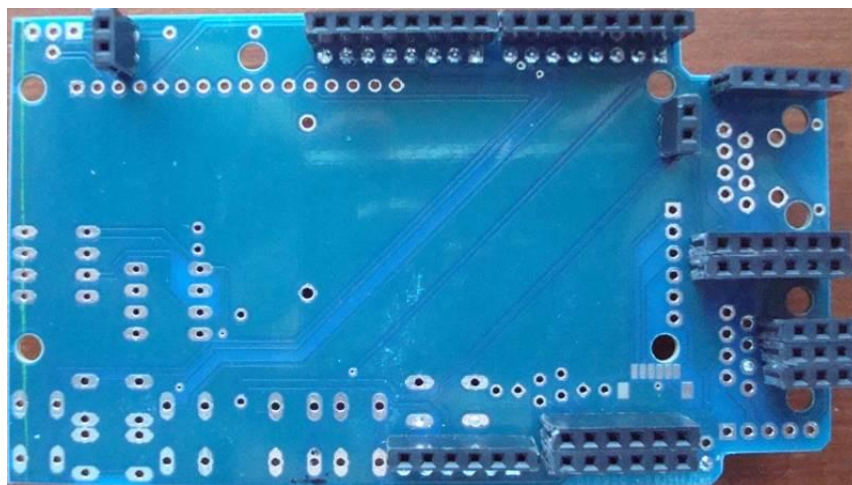


Fig. 10-6 Female pins are soldered on. Please solder a few rows at a time.

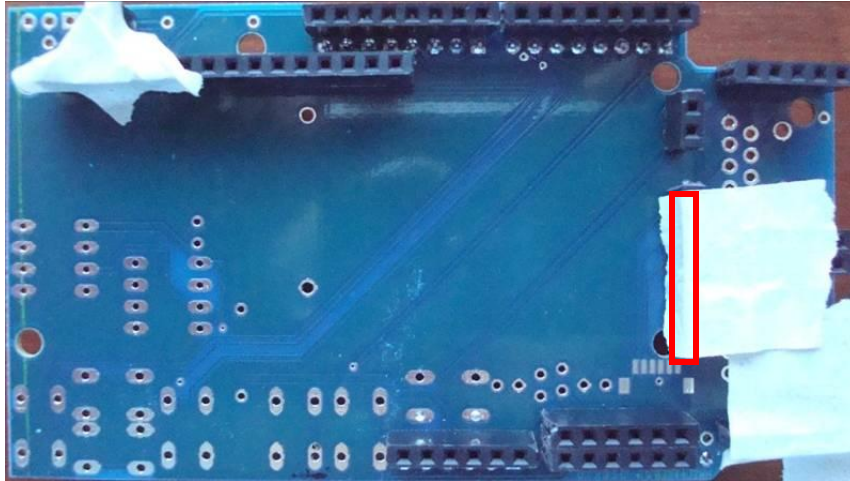


Fig. 10-7 Secure LCD female pins (top) and more female pins (red is GPS).

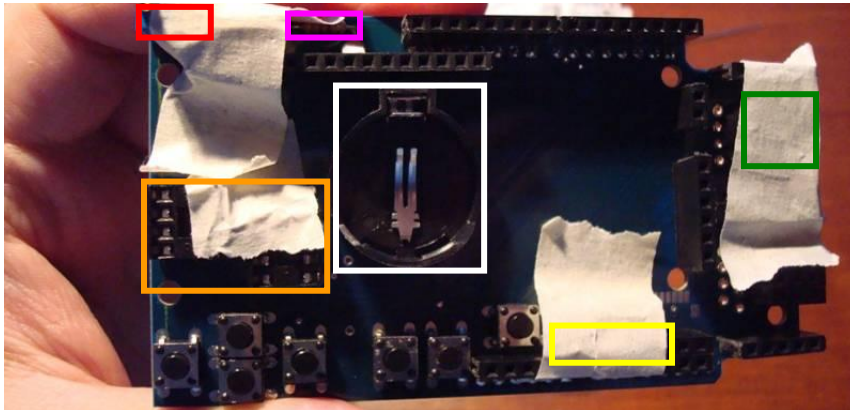


Fig. 10-8 Tape potentiometer, Dip sockets, LEDs, buzzer, resistors. Battery holder has double sticky tape in the back. Push the buttons in.

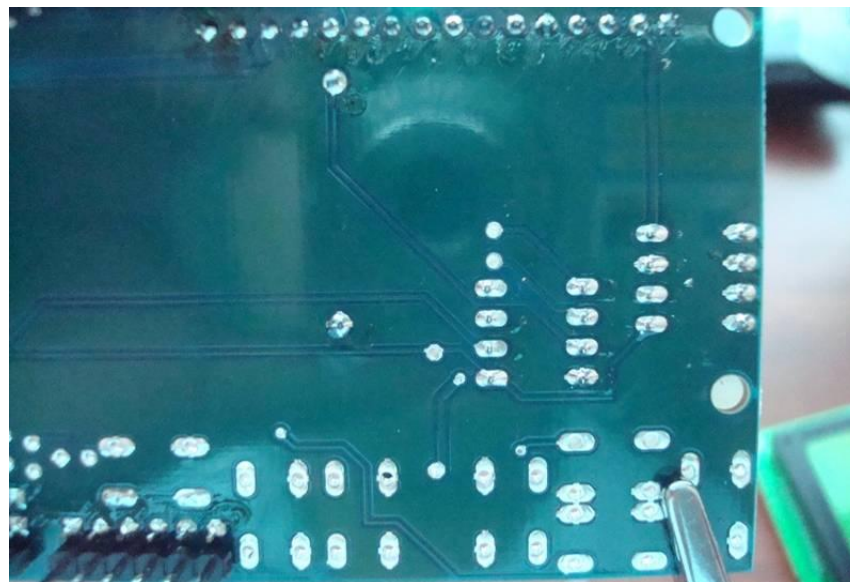


Fig. 10-9 Everything secured previously was soldered on (back of PCB).

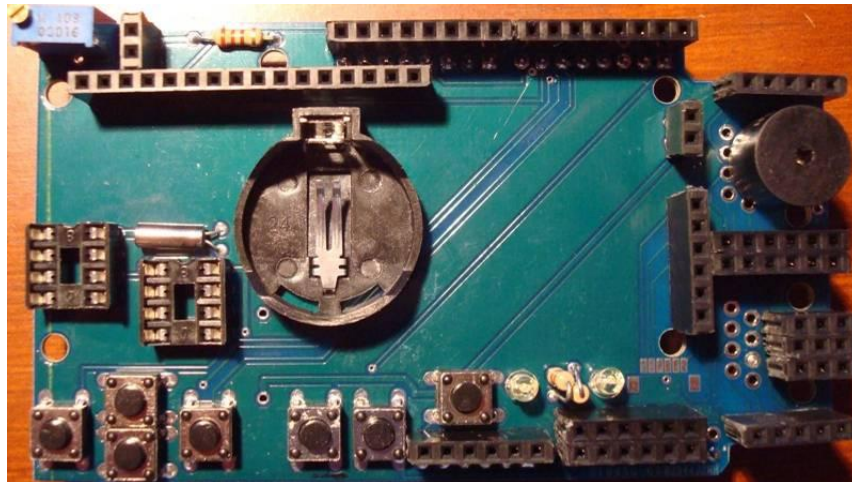


Fig. 10-10 Completed shield.

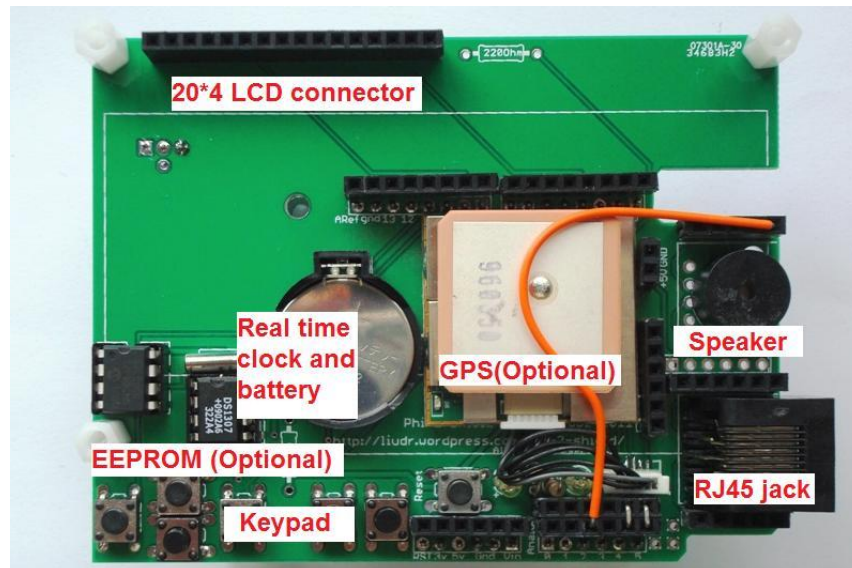


Fig. 10-11 Completed 20X4 shield. The RJ45 jack is not included in the kit.

## 11. Testing the shield:

Now that you're done, download the testing code to test out all functions:

[http://code.google.com/p/phi-prompt-user-interface-library/downloads/detail?name=Phi\\_2\\_testing\\_v2%2009242011.zip&can=2&q=#makechanges](http://code.google.com/p/phi-prompt-user-interface-library/downloads/detail?name=Phi_2_testing_v2%2009242011.zip&can=2&q=#makechanges)

1. You need to set the clock before you can use the code. Uncomment this line "#define setRTC" in the main program.
2. Update the time in setup() to the current time.
3. Upload the sketch to Arduino so the setup() will set the time.
4. Then comment the line "#define setRTC" and recompile and upload to Arduino.

5. If you don't do step 1, you will be stuck on the clock test. If you don't do 4, the clock will be reset every time Arduino resets.
6. Study Fig. 14-8. Make sure you jump the I2C connections. Jump buzzer pin to analog 2.
7. Follow the steps shown on the LCD. First turn the potentiometer all the way one way to get best contrast. If not turn all the way the other way.
8. Test all buttons per direction.
9. Test buzzer. While the buzzer is buzzing, disconnect the jumper to buzzer and connect to the two LEDs to test them as well. See Fig. 14-1 and 14-8 for LED connectors.
10. Then the real time clock is tested. After that if you have EEPROM it will be tested as well but if not, the program hangs at that test, no big deal.

Download more codes at <http://liudr.wordpress.com/phi-2shield/>. Use them as templates and develop your project!

## 12. Software:

The family of Phi shields and add-on hardware comes with the most extensive software support in Arduino-compatible hardware, ranging from code testing functions to fully-working project codes, to code templates to kick start your projects, and code snippets to quickly add functions to your existing codes. Although all codes may be modified and adapted to work on similar shields, the codes are optimized for the Phi shields.

What's included?

- Phi\_prompt: interactive user interface building function functions to build your LCD menu and interactions: [http://liudr.wordpress.com/libraries/phi\\_prompt/](http://liudr.wordpress.com/libraries/phi_prompt/)
- Phi\_interfaces: for buttons: [http://liudr.wordpress.com/libraries/phi\\_interfaces/](http://liudr.wordpress.com/libraries/phi_interfaces/)
- Complete project codes: see <http://liudr.wordpress.com/phi-2-shield/>
- Test code: see <http://liudr.wordpress.com/phi-2-shield/>

Phi\_prompt sample (download code at [http://liudr.wordpress.com/libraries/phi\\_prompt/](http://liudr.wordpress.com/libraries/phi_prompt/)):

```
• Spaghett→Rotelle
• Rotini • Fettucc
```

A list user can select from

```
Top Menu: →1/5
Collect data
```

A multiple-level menu

```
Number (0-20):
12
```

User can enter a number

```
File name:
DATA0012.TXT
```

User can enter text string

Fig. 12-1 Sample phi\_prompt lists and entries



Fig. 12-2 Alarm clock project code running on 20X4 shield

Code snippets:

To use the LCD:

First include this line in the beginning of your code:

```
#include <LiquidCrystal.h>  
LiquidCrystal lcd(8, 9, 7, 6, 2, 3); // This sets up the display.
```

Next, include the following in your setup () function:

```
lcd.begin(16, 2); // This initializes the display.
```

To print messages:

```
lcd.clear();  
lcd.setCursor(column, row);  
lcd.print(variable); or lcd.print(string);
```

If you want to use some of the pins that are used by the buttons, simply DON'T push those buttons and they won't bother you. Say you want to use pin 3 and pin 11 for two LEDs but they're used by the Left and Right buttons. You go ahead and hook the pins (female headers) to the LEDs (remember current-limiting resistors), then change the following codes from my samples so the sample code will not try to sense the buttons:

Locate these lines:

```
byte pins[]={btn_u,btn_d,btn_l,btn_r,btn_b,btn_a};  
char mapping[]={ 'U','D','L','R','B','A'};
```

Replace them with:

```
byte pins[]={btn_u,btn_d,,btn_b,btn_a};  
char mapping[]={'U','D','B','A'};
```

Now the buttons are deleted from the button\_array object, you can use their pins for other purposes.

Then in your next project you want the buttons back, just unhook the LEDs from the previous project and you're free to use the button again. Just undo the code changes described above.

## 13. Application notes:

### Using RJ45 for both powering and controlling/sensing a device

To reduce contact resistance on the crimped cable, you can use multiple wires in the RJ45 cable to pass power and gnd. Say you only need to control one servo motor, then you can chain 3 wires of the RJ45 together to pass power, 3 wires together to pass gnd, and 2 together to pass sensor.

If you plan to run both sensor and power down the same wire, you should put some filter capacitors between the power and gnd, to prevent the power to couple with sensor. In a project I did, I had to put a 1uF cap for my sonic ranger to work on 12ft of phone cord.

<http://liudr.wordpress.com/2011/02/12/arduino-parking-sensor/>



## 14. Pictures:

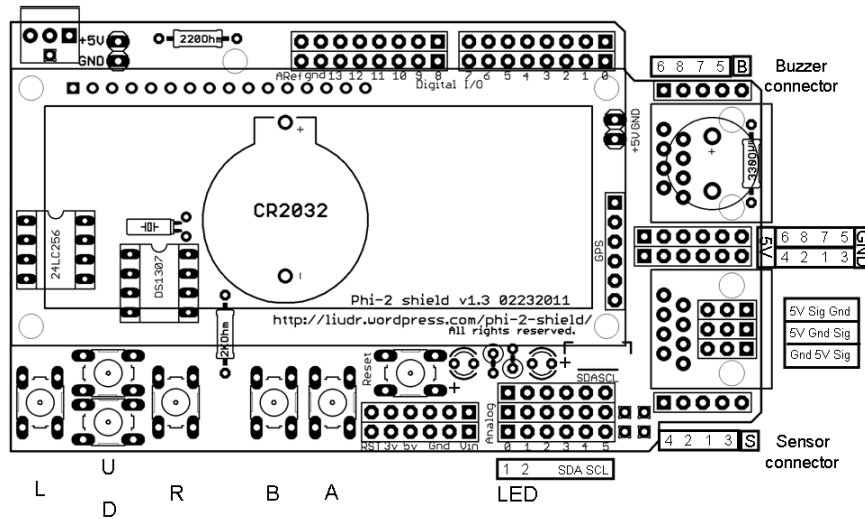


Fig. 14-1 Assembly and connection guide. Please print this out for your reference.

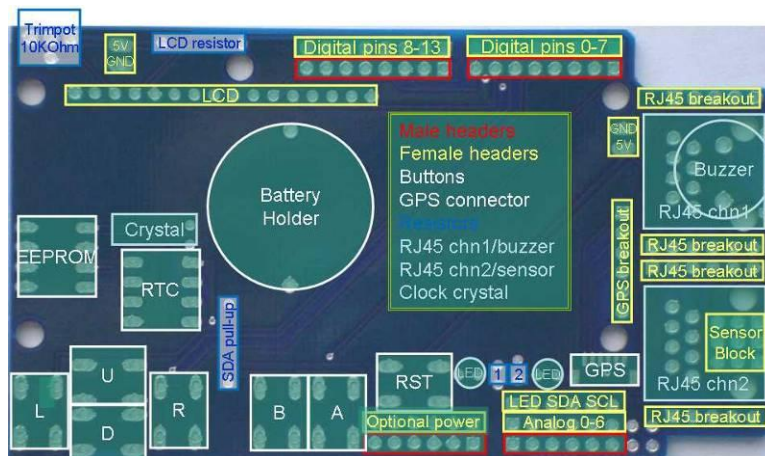


Fig. 14-2 Front side of the shield PCB with all parts marked.

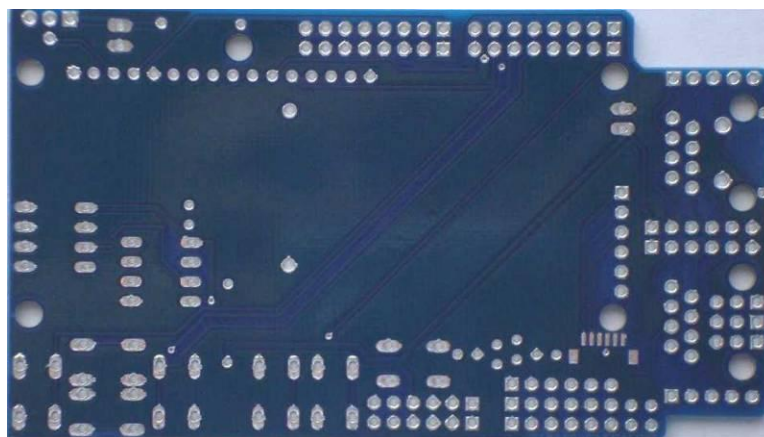


Fig. 14-3 Front side of a bare PCB

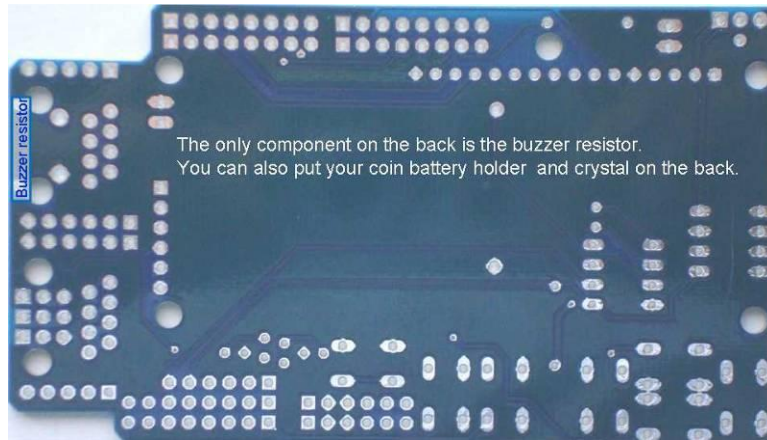


Fig. 14-4 Back side of the shield PCB.



Fig. 14-6 A completely assembled shield with buzzer and sensor block



Fig. 14-7 A completely assembled shield back side

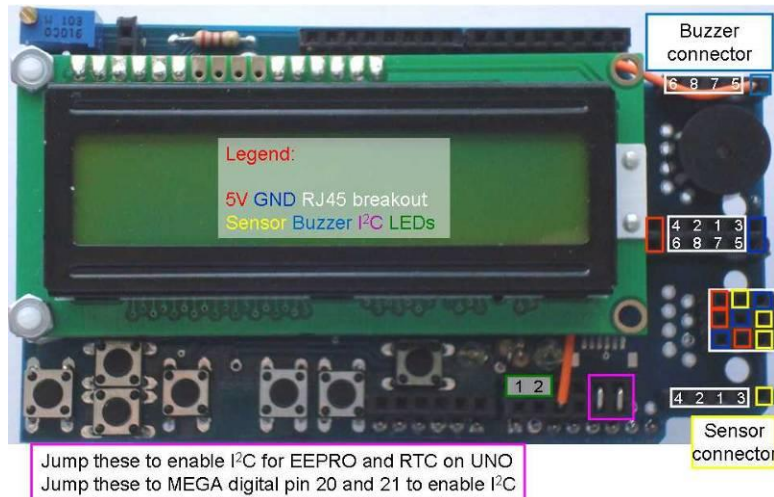


Fig. 14-8 A completely assembled shield with connection directions

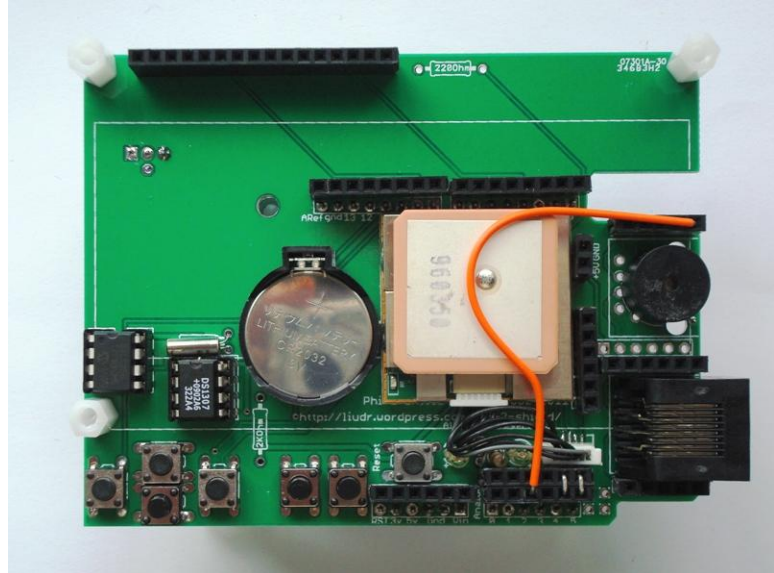


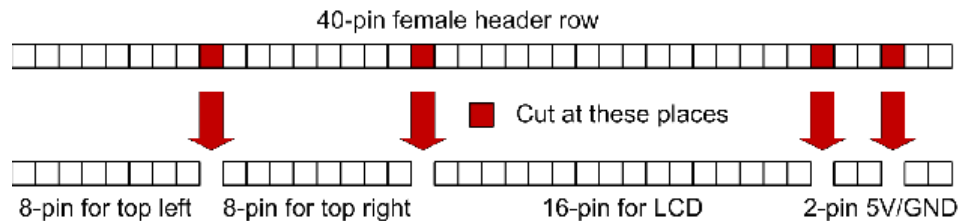
Fig. 14-9 A 20X4 shield with an EM 406A GPS module (LCD is removed to show GPS)

## 15. FAQ:

1. [How do I get the 2-pin, 6-pin, and 8-pin female headers pictured in your parts photo?](#)
2. [Where do I get an EEPROM, the GPS connector, and the GPS module?](#)
3. [Where to get a decent soldering iron for the assembling?](#)
4. [How do I solder the headers so they don't end up tilted or rotated?](#)
5. [How to choose LCD back light resistor?](#)
6. [How do I secure the LCD to the shield, besides pushing the headers together?](#)
7. [How do I test my assembled shield?](#)
8. [How do I make use of the RJ45 connectors?](#)
9. My real time clock (EEPROM) is not working!

### How do I get the 2-pin, 6-pin, and 8-pin female headers pictured in your parts photo?

I take one 40-pin female header (hope you bought two as required on the parts list), then I use a regular 45 degree cutter (\$2USD at a hardware store) to cut it to the size I need. Say I need a 6-pin female header, I take a row of female headers, clip on the 7th pin on both the pin side and the hole side, this destroys the 7th pin but after trimming the edges, I get a 6-pin header. I will then trim off the edge on the rest of the row of female pins, then cut one more time to get another header out. Remember, if you need a header with 6 pins, cut on the 7th pin so you get a complete 6-pin header. So for a 40-pin female header you will get say 8(keep)-1(break)-8(keep)-1(break)-16(keep for lcd)-1(break)-2(keep for 5V/GND)-1(break)-2(keep). Break indicates you destroy the pin to separate into the size you need. Then use the other 40-pin header row to get one more 8 and two 6. Here is an illustration:



### Where do I get an EEPROM, the GPS connector, and the GPS module?

Here is a list of the links to the vendor:

GPS connector: <http://www.sparkfun.com/products/579>

EEPROM: <http://www.sparkfun.com/products/525>

GPS module: <http://www.sparkfun.com/products/465>

### Where to get a decent soldering iron for the assembling?

I have used both a corded and a cordless iron. I like the cordless one slightly better because it is not as intimidating as the corded one.

Corded: RadioShack basic soldering set for \$7.99

Cordless: Weller BP860MP cordless dual-power soldering iron for less than \$20 depending on where you buy.

### How do I solder the headers so they don't end up tilted or rotated?

The best way is to tape the headers to the board using masking tape. You will wrap the header with tape only leaving one pin exposed for soldering. Then solder this one pin so the header will not move when you remove the tape. After that, solder all rest pins, starting from the opposite end to the first soldered pin.

### How to choose LCD back light resistor?

My basic rule of thumb: start with a large resistor (300Ohm), reduce it until you're satisfied with the brightness of the back light. Remember the back light doesn't have to glow visibly in a well-lit room. You just need it bright enough when all lights are out. This saves battery and life time of the back light.

Say you bought this from dipmicro (<http://www.dipmicro.com/store/LCD-1602A-B>), its spec sheet is here: (<http://www.gzliyedz.com/download/yjzpf/162-4.pdf>)

It says  $V=4.1$   $I=120\text{mA}$  about the back light. This current is kind of big but with calculation, you need 7.5Ohm resistor. I would use a larger value like 150 Ohms. I don't want my backlight to burn so bright that it reduces its life time. All you need is when the lights are turned off in the room, the back light makes the display totally visible.

<http://www.sparkfun.com/products/709>

This is very similar to the one dipmicro sells. The comments are quite helpful too.

### **How do I secure the LCD to the shield, besides pushing the headers together?**

There are three mounting holes on the shield that line up with the LCD mounting holes. Since the LCD mounting holes can vary a bit from manufacturer to manufacturer, I didn't put a specific screw size there. I would go to a local hardware store (like ace) and go to the screw/nut sector to hand pick some. You will need a standoff between the LCD and the shield. I ended up using M3 standoff and M3 screws. They're a bit too large but are available to me. I just screwed the screw through the board holes to open it up by just enough. Then screw down the standoff till it fits. Nothing fancy.

### **How do I test my assembled shield?**

First, you want to visually inspect everything to make sure there are not short circuits. Then carefully plug the shield into arduino while the arduino is not powered. Supply power to arduino. If arduino will not be recognized by the computer, then you have a short, which draws too much current that it trips the resettable fuse on arduino (bless the arduino designers for this feature). Check again. Remove excessive solder with some solder wick (radioshack) or solder sucker.

Once arduino powers up, load the test program from the list of codes on this page, test everything. If you don't see any message on the LCD, adjust the trimpot all the way to find the best spot for a good contrast. Without a proper setting on the pot, the LCD messages will not be visible.

### **How do I make use of the RJ45 connectors?**

I will add RJ45 breakout boards mentioned in section 8 to dipmicro.com for sale. Check on my blog and dipmicro.com for updates. You may save time and money if I make like 100 of them and have dipmicro.com offer them for sale, other than you making 2 of them at a higher cost.

### **My real time clock (EEPROM) is not working!**

Please make sure the chips are seated as described by the assembly and connection diagram, with their notches facing up. You may not swap the RTC and EEPROM chip sockets. Push the chips to make a good connection.