



Fish Out of Water - ATtiny board used with Arduino

by [marc.cryan](#) on April 16, 2013

Table of Contents

Fish Out of Water - ATtiny board used with Arduino	1
Intro: Fish Out of Water - ATtiny board used with Arduino	2
Step 1: Parts	3
Step 2: Assembled Board	3
Step 3: Solder - Battery holder	4
Step 4: Solder - Socket	4
Step 5: Solder - Header - JP3	5
Step 6: Solder - Headers, JP1,2	5
Step 7: Solder - Capacitor	5
Step 8: Attach to Arduino	6
Step 9: Program Arduino as ISP	7
Step 10: Insert ATtiny	7
Step 11: Program ATtiny	7
Step 12: Attach Blinky Fish	8
Step 13: Add Battery	9
Step 14: Arduino to ATtiny code conversion	9
Step 15: Schematic and PCB	9
File Downloads	9
Step 16: Design Files	9
File Downloads	9
Related Instructables	10
Advertisements	10



Author: marc.cryan

Married to Domestic_Engineer (but I call her Meghan).

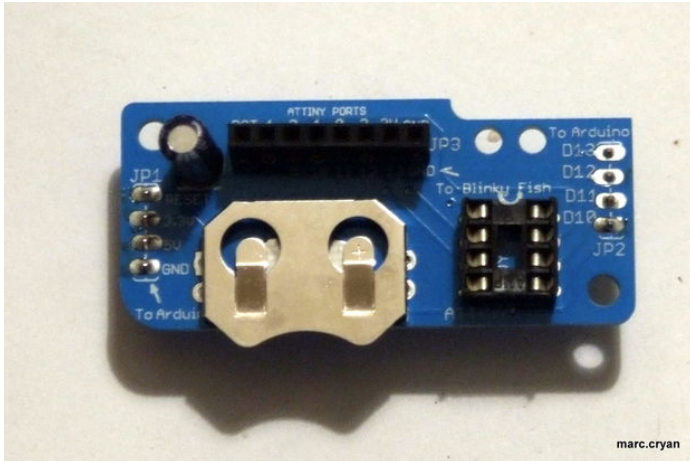
Intro: Fish Out of Water - ATtiny board used with Arduino

FishBowl is a small microcontroller board programmable using the Arduino software. It has a battery so you can then use it without the larger Arduino main board. It is like a tiny Arduino to-go.

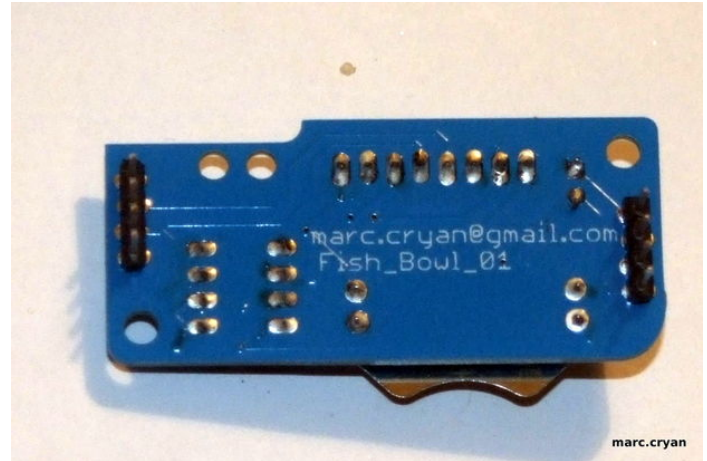
Build instructions for the Fish Out of Water Kit from the [Blinky Fish Kickstarter Project](#)

Steps:

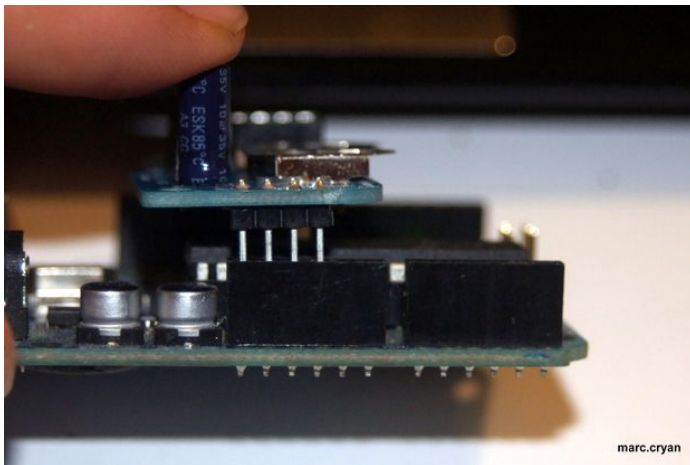
1. Assemble with soldering
2. Program an Arduino as a programmer (optional, kit comes with pre-programmed ATtiny)
3. Program ATtiny microcontroller with the Arduino (optional, for same reason)
4. Attach Blinky Fish



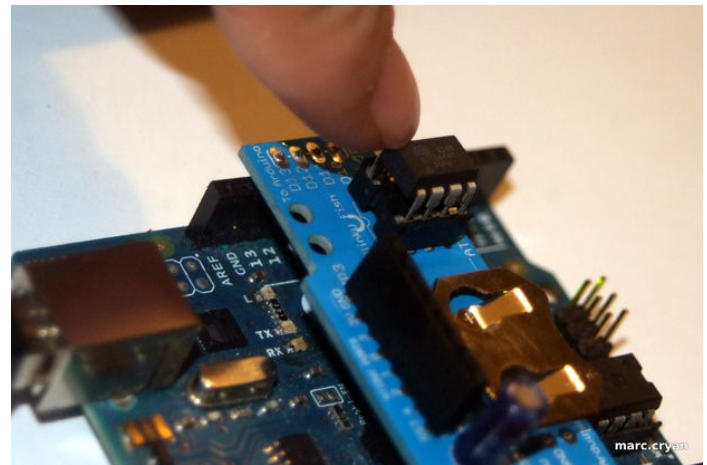
marc.cryan



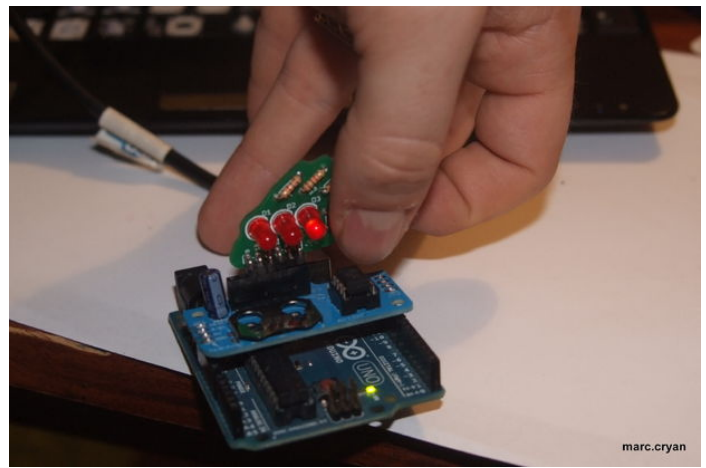
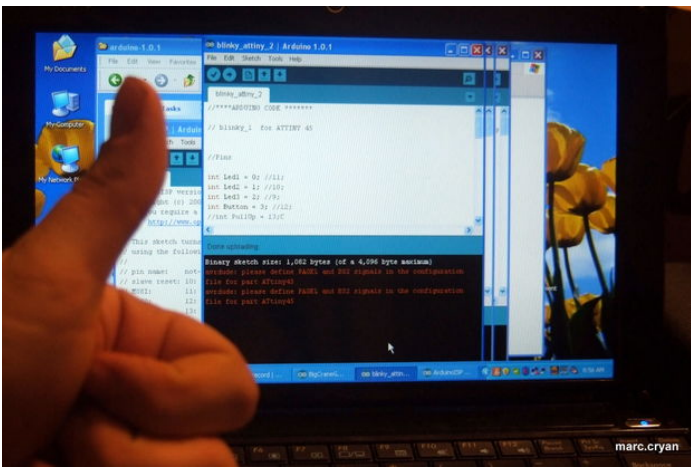
marc.cryan



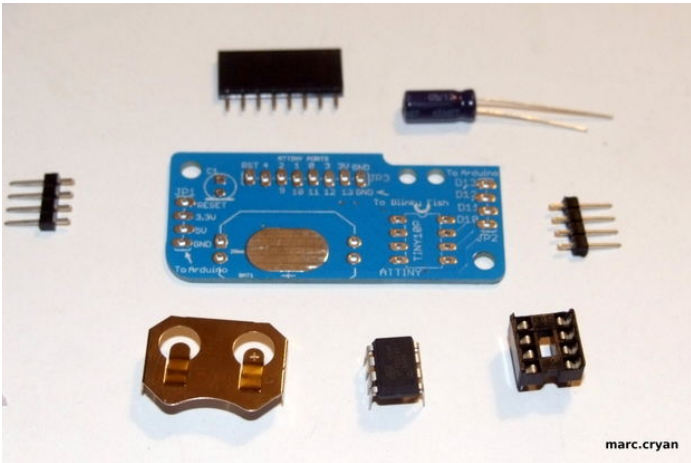
marc.cryan



marc.cryan

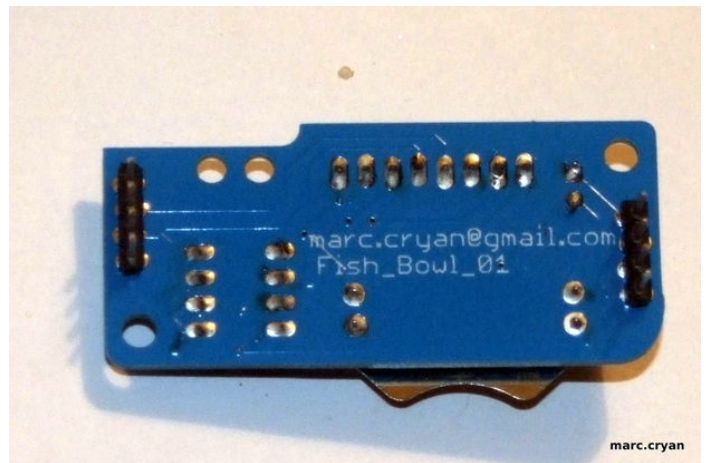
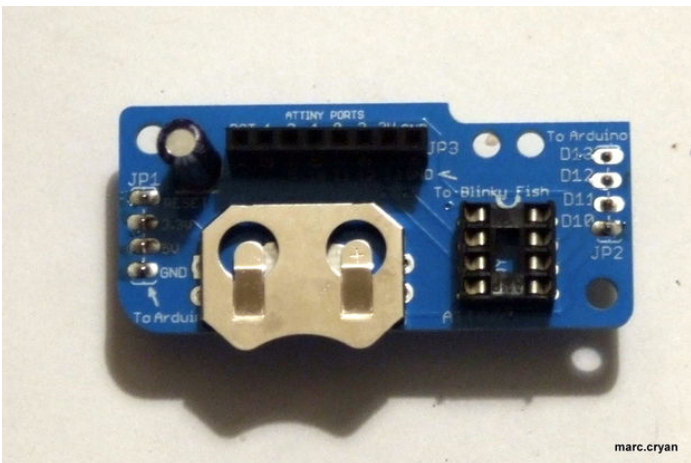


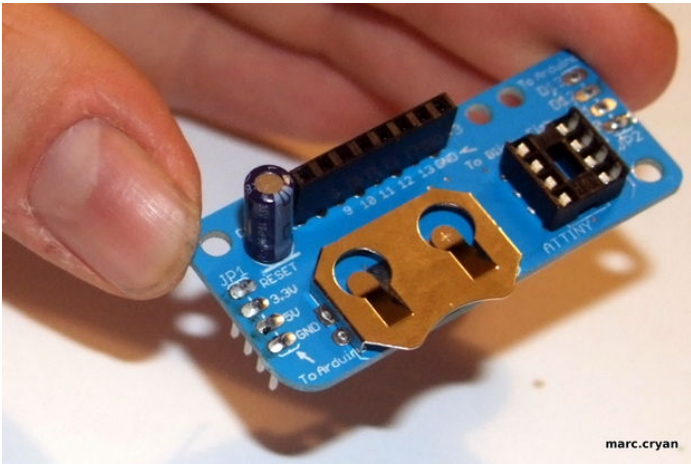
Step 1: Parts



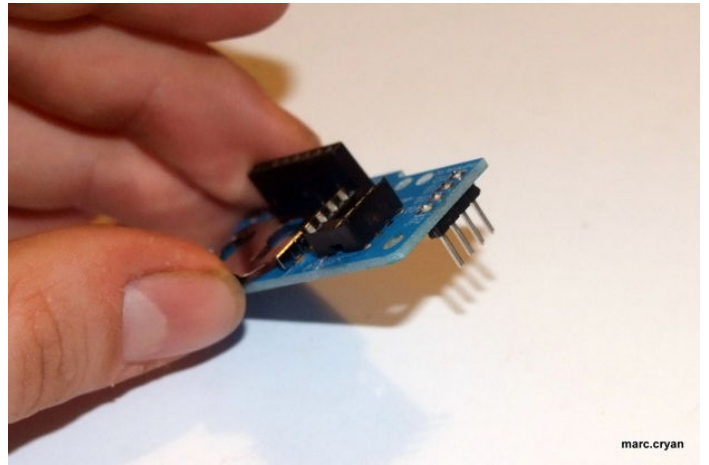
Step 2: Assembled Board

Here is the final assembly. This might be all you need. Notice the line on the capacitor goes towards the line on the board. The header pins go on the bottom of the board. The header sockets go on the top. There is a socket to hold the ATtiny chip. The battery clip hold a CR2016 button cell battery. It can be modified to hold the larger CR2032 battery.



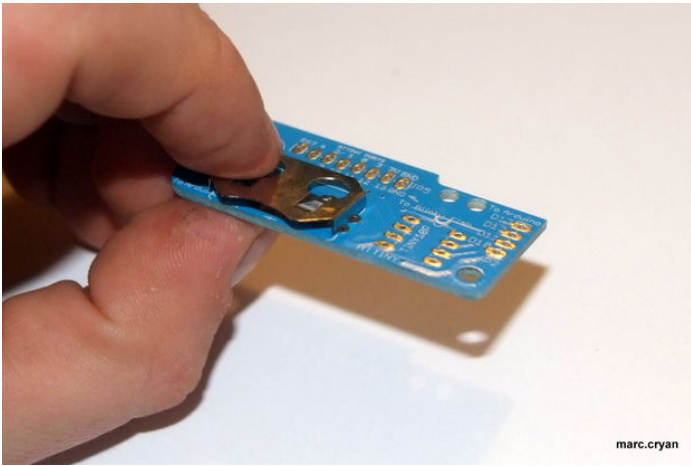


marc.cryan

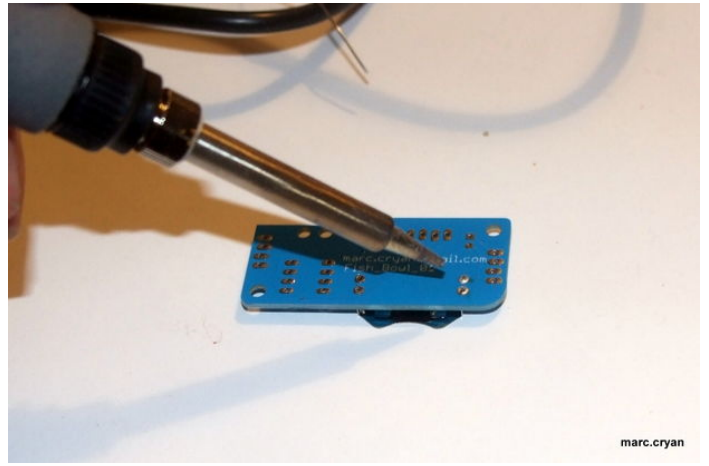


marc.cryan

Step 3: Solder - Battery holder

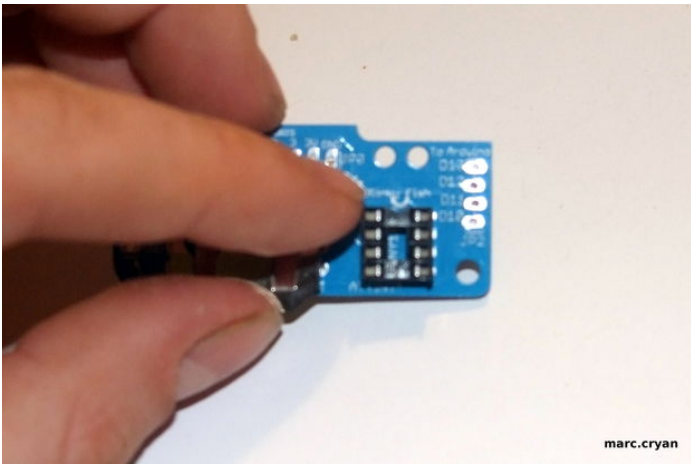


marc.cryan

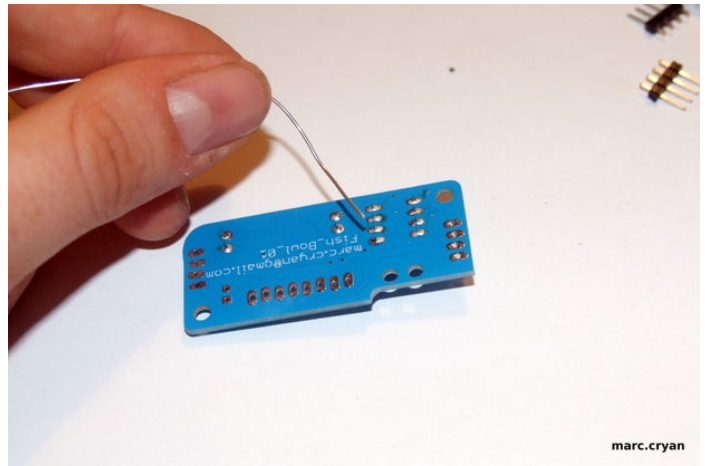


marc.cryan

Step 4: Solder - Socket



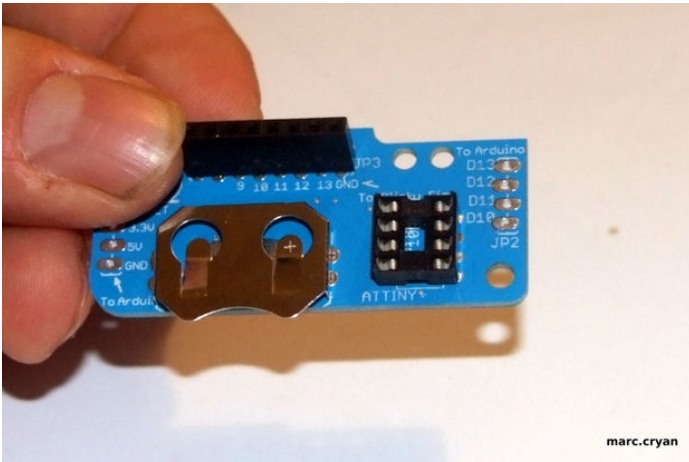
marc.cryan



marc.cryan

Step 5: Solder - Header - JP3

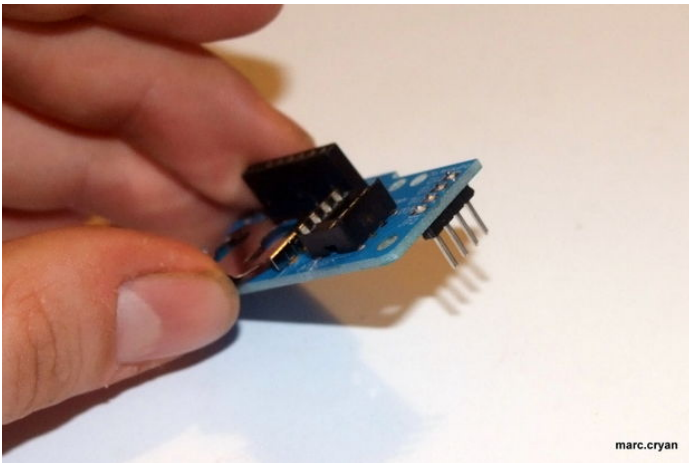
Sockets go on top of board.



marc.cryan

Step 6: Solder - Headers, JP1,2

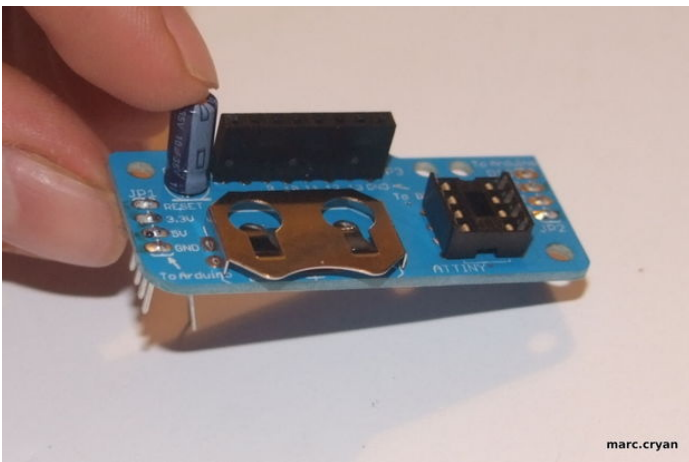
Pins go on the bottom of board. If you do not plan to re-program the ATtiny chip then you can leave these off. You could also try leaving these un-soldered, they should stay in place will enough to program with the Arduino and then be removed after programming.



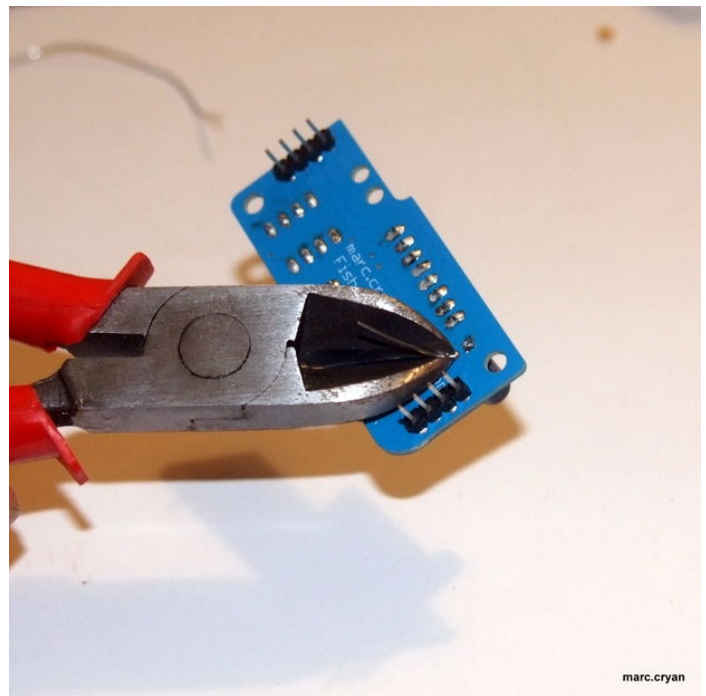
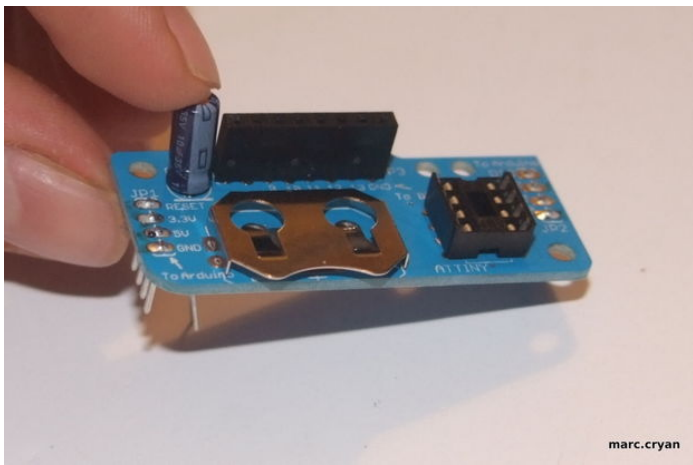
marc.cryan

Step 7: Solder - Capacitor

Line goes towards the line.



marc.cryan

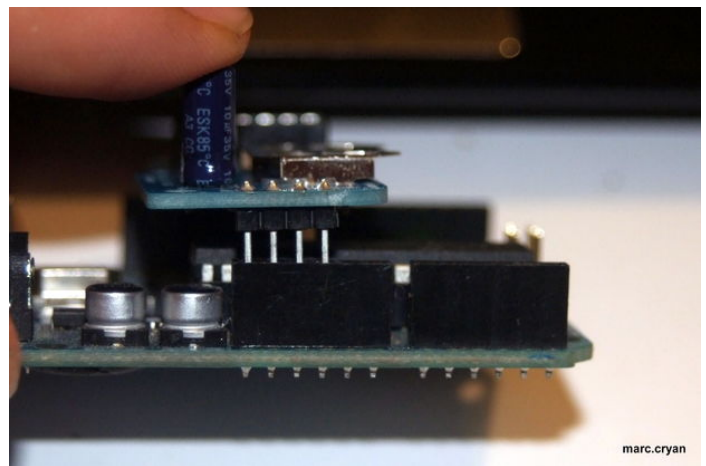
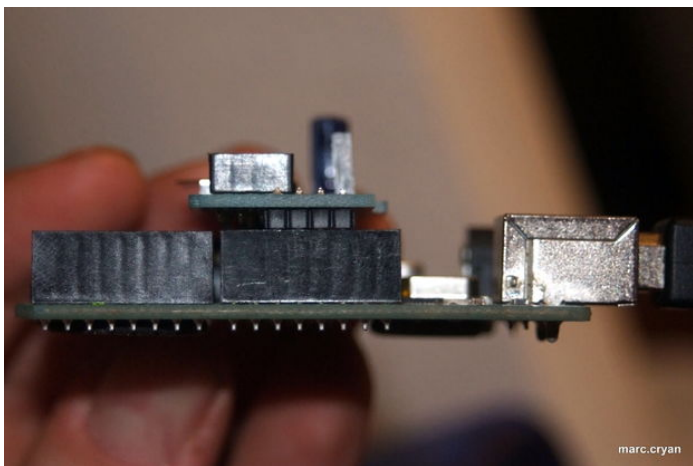
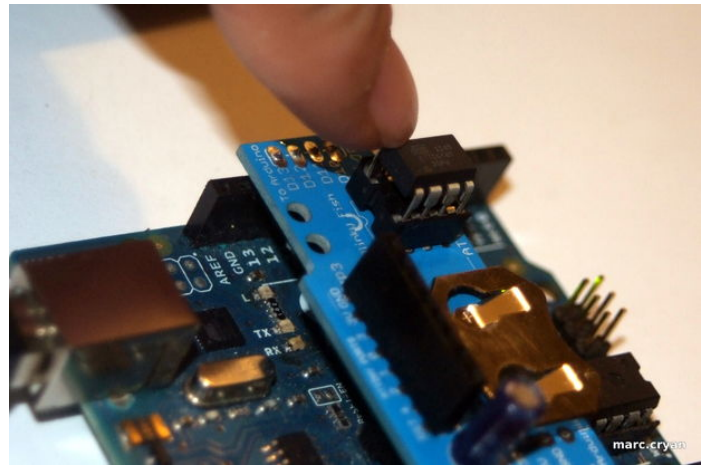
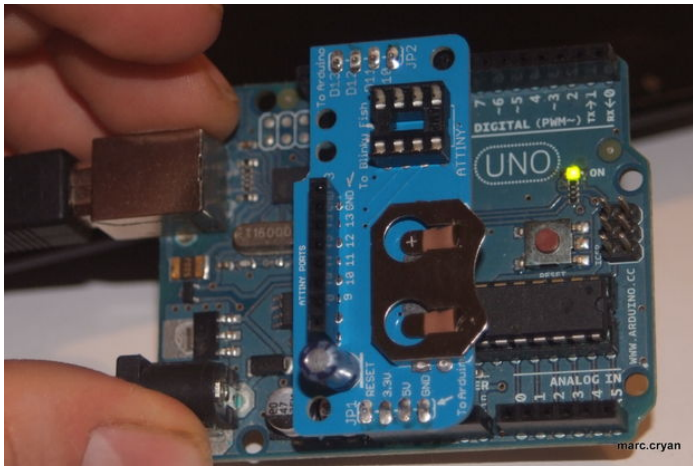


Step 8: Attach to Arduino

These are the steps to program the ATtiny board using the Arduino main board.

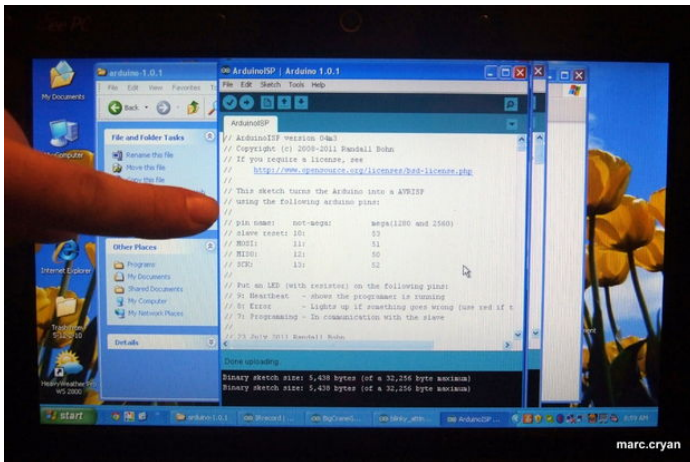
You can skip this if you have the March 2013 Kickstarter kit, the chip is pre-programmed.

The next few steps are derived from this - <http://hlt.media.mit.edu/?p=1695>



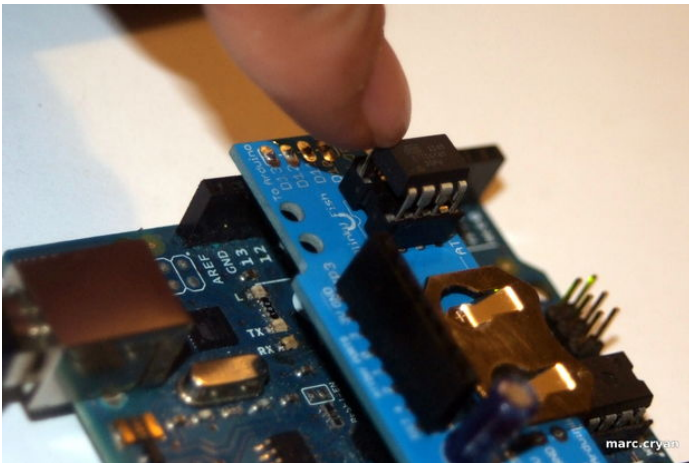
Step 9: Program Arduino as ISP

In Arduino software version 1.0.1, go to Examples and open ArduinoISP. Download this program to the Arduino board.



Step 10: Insert ATtiny

Insert the ATtiny into the socket.



Step 11: Program ATtiny

1. Open a new sketch and copy the code below.
2. Select Tools -> Programmer -> Arduino as ISP
3. Select Tools -> Board -> ATtiny45 1MHz clock
4. Download to the ATtiny, lights will blink (look through the 2 holes), there will an error message (2 of the same message) but that is okay.

```
/** ARDUINO CODE **/
```

```
// blinky_2 for ATTINY 45
```

```
// Pins
```

```
int Led1 = 0; //11;
int Led2 = 1; //10;
int Led3 = 2; //9;
int Button = 3; //12;
//int PullUp = 13;
```

```
int ButtonState = 0;
int oldButtonState = 0;
```

```
void setup() {
  pinMode(Led1, OUTPUT);
  pinMode(Led2, OUTPUT);
  pinMode(Led3, OUTPUT);
  pinMode(Button, INPUT);
  //pinMode(PullUp, OUTPUT);
  digitalWrite(PullUp, 1);
}
```

```
void loop() {
```

```
  ButtonState = digitalRead(Button);
```

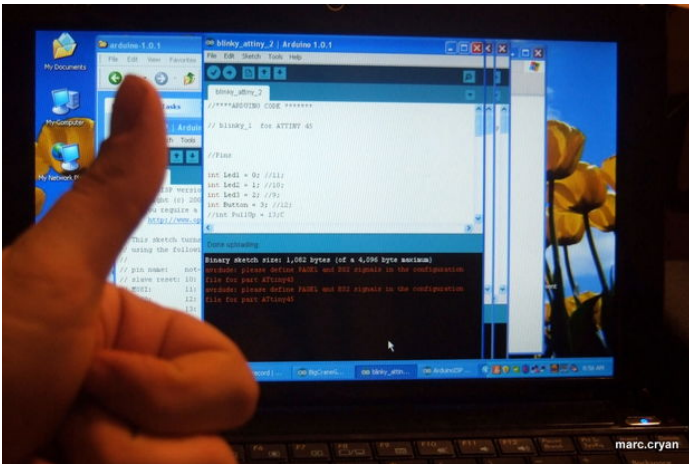
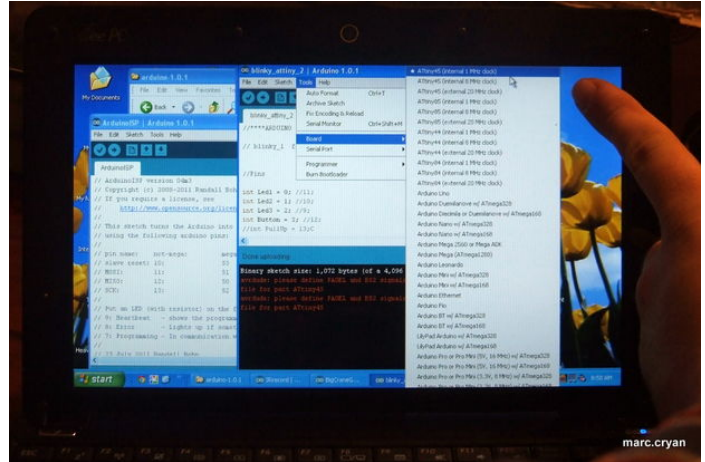
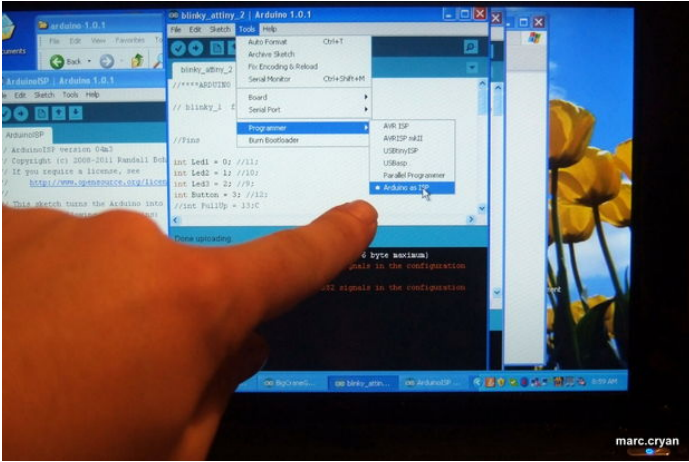
```
  if (ButtonState != oldButtonState) {
    if (ButtonState == HIGH) {
      digitalWrite(Led1, 1);
```

<http://www.instructables.com/id/Fish-Out-of-Water-ATtiny-board-used-with-Arduino/>

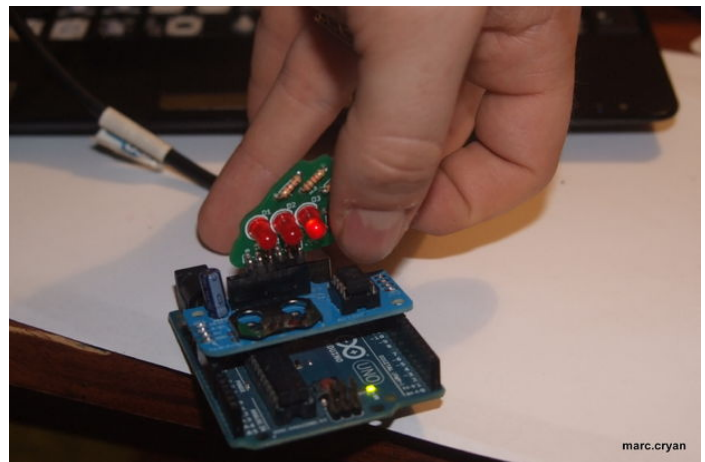
```

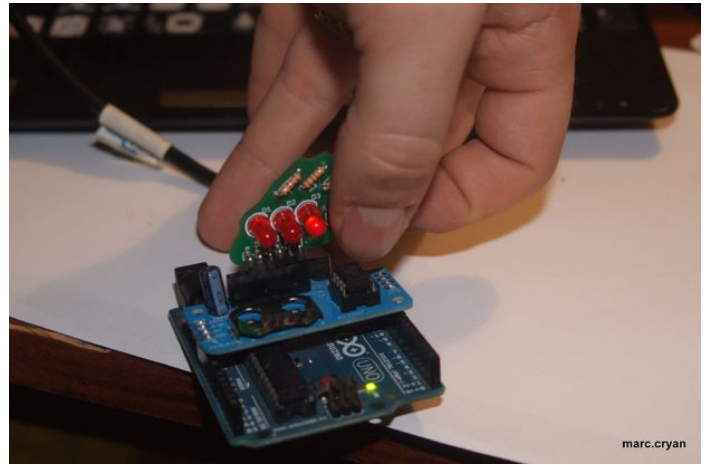
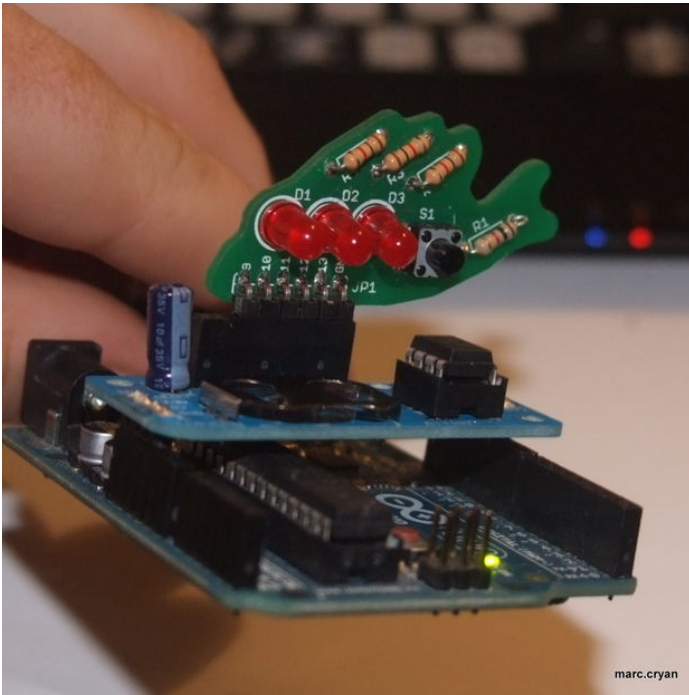
delay(300);
digitalWrite(Led2,1);
delay(300);
digitalWrite(Led3,1);
delay(300);
}
else{
digitalWrite(Led3,0);
delay(300);
digitalWrite(Led2,0);
delay(300);
digitalWrite(Led1,0);
delay(300);
}
oldButtonState = ButtonState;
}
}

```



Step 12: Attach Blinky Fish





Step 13: Add Battery

Step 14: Arduino to ATtiny code conversion

To make the Arduino code work the the ATtiny, we changed these pins.

```
Aduino -- ATtiny  
11 -- 0  
10 -- 1  
9 -- 2  
12 -- 3
```

Step 15: Schematic and PCB

File Downloads



Attiny_3_toFAB_PCB.pdf (13 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Attiny_3_toFAB_PCB.pdf']



Attiny_3_toFAB.pdf (13 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Attiny_3_toFAB.pdf']

Step 16: Design Files

File Downloads



ATTINY_GERBER_X5.zip (10 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'ATTINY_GERBER_X5.zip']



Attiny_3_toFAB.sch (114 KB)

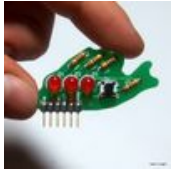
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Attiny_3_toFAB.sch']



Attiny_3_toFAB.brd (41 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Attiny_3_toFAB.brd']

Related Instructables



Blinky Fish by
marc.cryan



**Activate LED
with button --
Arduino
Example Video**
(video) by
marc.cryan



**Blinky Fish -
Kickstarter
Project** by
marc.cryan



**Chipper Board -
ATtiny
Programming
Shield** by
Hammock Boy



**Wendell the
Robot** by
marc.cryan



**Stripboard
Arduino shield
for
programming
ATtiny45 and
ATtiny85** by
nikkipugh