Engaging middle school and high school students in STEM through a programming, design, and soldering workshop

Prof. Kevin P Pintong, Oregon Institute of Technology

Kevin Pintong is an assistant professor at Oregon Institute of Technology in Klamath Falls, Oregon.
Engaging middle school and high school students in STEM through a programming, design, and soldering workshop

Kevin Pintong, Oregon Institute of Technology
1. Introduction

The Owlet board is a development kit designed to encourage middle school and high school students to pursue STEM majors. It was developed as a community outreach and marketing tool for Oregon Tech’s Computer Systems Engineering Technology (CSET) department with support from Oregon Tech Commission on College Teaching and Microchip Corporation. This board was designed to be a recruiting tool for potential students, and to engage the community in STEM. Participants can build the board, program the board, and then take the board home for further design and exploration. Many activities targeting younger STEM students focused on physical (Lego Mindstorms or similar) or virtual (coding only). This project was an attempt to incorporate both hardware and software based concepts into one tool.

In Spring of 2015, the Owlet board went under development after funding approvals. Sufficient parts for 200-300 boards were ordered, and documentation was developed. In Fall 2015 the first workshop on how to solder and program the Owlet board was run. The workshop was attended by prospective students from Ponderosa High School and Rogue Community college as well as current Oregon Tech students and staff. Although there were issues and concerns regarding the availability of soldering irons and lack of organization, participants felt that the experience was overwhelmingly positive.

2. Workshop design
The intent of the workshop is to provide a brief overview to engage students. It is not intended to provide an in-depth understanding of circuit design and programming. The intended design of the workshop is to cover three activities in order to target students who might be interested in Computer Science, Computer Engineering, and Electrical Engineering. The three activities of concern are:

- Understanding circuits and components
- Placing and soldering components
- Programming the board using software

The following were the intended learning outcomes of the workshop.

1. Participants will learn how to read a bill of materials and schematic.
2. Participants will learn how to identify and orient circuit components such as resistors, capacitors, LEDs, and microcontrollers.
3. Participants will learn how to solder through-hole components.
4. Participants will learn how to use reference designators on the board, bill of materials, and schematic to place components.
5. Participants will learn how to program the board using the MPIDE interface.
6. Participants will learn how to blink an LED on the board at different speeds.

3. Board design
When designing the development kit, we had the following goals:
• Easy for students to solder
• Easy to program
• Low cost

3.1 Easy for students to solder
Since this is the first time students are seeing these components, it is important that components are mainly through-hole to be easily soldered. It is also important to reduce the number of pins soldered to complete the workshop in a timely manner. This means reducing the number of components to a minimum.

3.2 Easy to program
Since this is the first time participants may be programming, C programming language would be too difficult. Something such as Wiring (Arduino) is more suited for a first time participant. The board should also be programmable via a built-in USB bootloader so participants can take it home and continue to explore the board. Requiring an additional expensive debugger would mean students could not continue to play with the board at home.

Note that microprocessors that have built-in bootloaders are sometimes more difficult to interact with since the bootloader must share code space with the user code. This means that additional steps must be taken in order to program the board.

3.3 Low cost
In order to reach more prospective students and utilize the development board as an advertising tool, it is important that participants can keep the board at the end of the workshop.

3.4 Design and cost reduction strategies
The board design was performed in KiCAD. In order to save costs, the board was reduced to 5 cm x 10 cm and 2-layer. PCB vendors typically provide a substantial discount for designing within a 5x10cm or 10x10 cm size limit. This size fits within the PCB vendor’s low-cost option.

Another cost savings is to utilize a microcontroller with open-source USB bootloader. Removing programming circuitry drastically reduces the number of parts required. The board was based around the PIC32MX250F128B, which Microchip provides pre-programmed with USB bootloader firmware [1]. Note that the ATTINY85 and ATTINY84 from Atmel also have VUSB bootloader firmware available, and may be a better choice considering the lower pin count [2].

The bootloader code and schematic is taken from the ChipKit DIY schematic [3], but the actual PCB layout was designed to be in the shape of an Owl. Other alternatives include using the opensource DigiSpark schematic and bootloader code as a starting point [4].

In order to reduce costs, the USB connector was removed. An on-board USB connector was designed into the board as shown below. Unfortunately, the PCB based USB connector requires gold plating and 2.0-2.4 mm thick boards, which is an additional charge from most PCB vendors. At the time of design (2015), the cost differential between tin plated standard 1.6mm thickness boards and gold plated 2.0 mm thickness boards was low. This cost has increased since 2015 and it may be better to just use a discrete USB connector, which can cost $0.46.
The schematic, board layout, and bill of materials are presented below.

Figure 1 Schematic for board design

Figure 2 PCB layout

Figure 3 Actual manufactured PCB
3.5 Bill of Materials and cost
The bill of materials includes 21 components with a total of 87 through hole pins and 4 surface mount pins to solder. Note the emphasis of easy to solder through-hole components in the design.

The printed circuit boards were ordered from Elecrow in Shenzhen, China. In quantities of 200, the per PCB cost was $1.13 [3]. Note that there was an additional charge for gold plating and additional PCB thickness, which are required if using the built-in USB connector.

Other parts were ordered from Ebay and Amazon vendors from China. The microprocessor was donated from Microchip, so there was no charge in this project.

<table>
<thead>
<tr>
<th>Part</th>
<th>Footprint</th>
<th>Reference(s)</th>
<th>Quantity</th>
<th>Approx Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 MHz Crystal</td>
<td>HC-49</td>
<td>Y1</td>
<td>1</td>
<td>0.15</td>
</tr>
<tr>
<td>3mm LED</td>
<td>3mm LED</td>
<td>D1, D2</td>
<td>2</td>
<td>0.10</td>
</tr>
<tr>
<td>330 ohm resistor</td>
<td>Through-hole</td>
<td>R5,R6</td>
<td>2</td>
<td>0.06</td>
</tr>
<tr>
<td>10k ohm resistor</td>
<td>Through-hole</td>
<td>R3,R4,R7, R8</td>
<td>4</td>
<td>0.06</td>
</tr>
<tr>
<td>36 pF capacitor</td>
<td>Through-hole</td>
<td>C1,C2</td>
<td>2</td>
<td>0.10</td>
</tr>
<tr>
<td>1 uF capacitor</td>
<td>Through-hole</td>
<td>C4,C5</td>
<td>2</td>
<td>0.10</td>
</tr>
<tr>
<td>10 uF capacitor</td>
<td>Through-hole</td>
<td>C3</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>Push button</td>
<td>Through-hole</td>
<td>SW1,SW2,SW3</td>
<td>3</td>
<td>0.04</td>
</tr>
<tr>
<td>PIC32 Microprocessor (PIC32MX250F128)</td>
<td>DIP-28</td>
<td>U3</td>
<td>1</td>
<td>4.05</td>
</tr>
<tr>
<td>1x10 0.1” header</td>
<td>0.1” header</td>
<td>P5,P6</td>
<td>2</td>
<td>0.10</td>
</tr>
<tr>
<td>3.3 volt linear regulator (AMS111733)</td>
<td>SOT-223</td>
<td>U2</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>PCB</td>
<td>PCB</td>
<td>PCB</td>
<td>1</td>
<td>1.13</td>
</tr>
</tbody>
</table>

Total cost excluding PIC32 microcontroller: $2.76
Total cost including PIC32 microcontroller: $6.81

4. Workshop implementation

4.1 Safety briefing
Participants were first briefed on safety and liability, and reminded that they will use soldering equipment, solder and flux. They were advised that what they would be working with would be hot and that solder and flux may contain hazardous substances such as lead or respiratory irritants. Participants were advised to wear safety glasses and use fume extractors to reduce exposure to solder fumes, as well as wash their hands after the workshop.
Additionally, participants were also reminded that Oregon Tech was not liable for injury stemming from participation in the workshop.

4.2 Parts
Components such as LEDs, oscillators, push buttons, resistors, capacitors, and the microprocessor were then explained to participants in brief detail. Participants were provided a list of components similar to Table 1 Bill of materials and costs. Additional emphasis was put on polarized components such as the light emitting diodes or the orientation of the microprocessor. Non-polarized capacitors were used for safety reasons.

4.3 Populate parts and solder
Participants then began to place all the components onto the board. Problems were encountered with the polarized components being placed in the wrong direction, particularly the LED. It is important to clearly notate the direction on the board, as well as explain to participants how the LED functions. It is also important to show participants how to use the tweezer to place the surface mount linear regulator. Below are some images and instructions from the workshop. Note that most participants were able to complete the board within 1-3 hours.
5 Program the board using Arduino environment.

Participants were briefed on how to setup the board. The software was pre-installed on lab machines but instructions were provided on how to install the drivers and MPIDE on their home computer. MPIDE was a clone of Arduino for PIC microcontrollers [4]. Since the release of Arduino 1.6, the preferred method is to use Arduino IDE and install the PICkit support library [4]. Since there was limited time, participants were briefly taught how to use MPIDE. First, participants went to the device manager to figure out the COM port of the device. Then the appropriate board, ‘DP32’ was selected in MPIDE.
Participants were told to load the ‘blink’ example and change the delay interval to blink it faster or slower. This process also showed how to load the program to the board.

```
// This example code is in the public domain.
/*
   #if !defined(PIN_LED1)
   #define PIN_LED1 13
#endif

void setup() {
  // initialize the digital pin as an output.
  pinMode(PIN_LED1, OUTPUT);
}

void loop() {
  digitalWrite(PIN_LED1, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(PIN_LED1, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

Figure 12 Blink example in MPIDE
6. Photos of participants for workshops

Below are images of the workshop participants working on their PCBs, as well as an image of the group completing their PCBs.

Figure 13 Two participants soldering

Figure 14 Participants finished with workshop
7. Lessons Learned and solutions to problems

The following are lessons that were learned during development of the board or implementation of the workshop.

7.1 Logistics

- There were too many component pins to solder. In order to save time in the future, another microcontroller such as the ATTINY84 could be chosen. This would reduce the number of pins from 28 to 14. By reducing pin count, time to solder can also be reduced.

- There was too much to do in one session. Future workshops should either be simplified or broken out into two workshops. One workshop could be held on circuit design and how to solder the board. The second workshop could be dedicated to learning how to program the board with more complex examples.

- Limit the number of participants to a reasonable number. Ideally, there will be lab assistants to help facilitate the workshop. A ratio of four participants to one teaching assistant is recommended.

- Parts should be prepared and kitted in advanced to save time during the workshops. Originally, the parts were not kitted, but distributing the parts themselves served to use up 15-20 minutes.

7.2 Design

- The current design costs too much to produce, and a lower cost microcontroller such as the ATTINY84/85 could be used.

- Future workshops should avoid surface mount components in order to save time. Also note that newer items such as USB micro connectors are far too difficult for first time participants.

- Polarized components need to be more clearly notated on the board, since participants made mistakes while soldering. Participants should also be reminded of the orientation of LEDs. Be prepared to remove these using a desoldering pump or wick.

7.3 Pedagogy

- There was insufficient explanation of how circuits and software worked. In the future, the tutorial should be updated to include much more information on circuit design and software examples.
Better explanation of how to program the boards is required. The PIC32 bootloader was difficult for participants to interact with, as the serial COM port sometimes changed, and participants would not program in time before the bootloader timed out.

Some participants were done with soldering in one hour, whereas others took up to three hours. This presents a problem since it is difficult to synchronize the workshop with all participants. Documentation should be developed in order to provide the early participants more tasks.

Development of additional software content beyond the basic LED blinking tutorial is necessary.

8. Additional learning outcomes

Although this workshop is designed to be basic and engage participants, it could include more circuit design and software. If the soldering activity can be reduced in time, the following objectives can be added that increase circuit design and software understanding.

1. Participants will understand Ohm’s Law.
2. Participants will learn how to use the button on the Owlet board to toggle an LED.
3. Participants will learn the effect Ohm’s law has on the brightness of the LED.
4. Participants will use pulse width modulation to slowly increase and decrease the brightness of the LED.

9. Conclusion

The workshop continues to be popular for participants, and basic workshop learning objectives are met. One major problem with this workshop is the overemphasis of soldering and assembly. More emphasis needs to be placed on understanding the circuit, understanding how the components work, and how to program the board. In order to correct this, future board designs can be simplified to reduce soldering time, and the documentation provided to participants can include more description of Ohms Law and circuit components. Many participants said they enjoyed the soldering workshop but wished they could have had more experience in programming the board. Additional software tutorials such as a pulse width modulation example can be added to the documentation provided to participants.

A future version of the workshop is in the process of being developed and refined. The board will be redesigned and simplified using a lower cost microprocessor. Additional explanations of circuit diagrams, components, and additional software tutorials will be added to the workshop.
References


