Project Name: The Charming Rod

Team Name: Stickle

Team Members:
Name: Jenna Stallings
Name: Wesley Hickle

Project Abstract:
Our project will consist of building a “magic” wand that can be used as a universal infrared remote control. An accelerometer will be utilized to gain movement information which will then be processed by a micro-controller. In this case, the micro-controller is the low-power MSP430. The micro-controller will process the incoming accelerometer information into an infrared signal that can be sent to the television receiver using a transmitter. A wand shaped casing will be made that can contain all of the above mentioned components. However, if space does not allow, the accelerometer will be encased and an external processing and transmitting unit will be implemented. Concerning operations, the micro-controller will process the information from the accelerometer to determine the pattern of movement being made. Based on information stored in the memory, the micro-controller will send out information via an infrared transmitter to do basic controls for the television. These basic controls include power on/off, volume adjustment, and changing of the channel.
Table of Contents

Introduction: .......................................................................................................................... 3
Technical Objectives: ........................................................................................................... 3
Project Architecture: .......................................................................................................... 3
Manufactured Product design ............................................................................................... 4
Cost Objectives: .................................................................................................................. 4
Membership Responsibility Table ....................................................................................... 5
References: ........................................................................................................................... 5
Appendix A: Altium Schematic ........................................................................................... 6
Appendix B: Altium PCB ....................................................................................................... 6
Appendix C: Code .................................................................................................................... 6

List of Figures and Tables

Figure 1: Block diagram of The Charming ........................................................................... 3
Figure 2: Picture of final product ......................................................................................... 4
Table 1: Bill of Materials ..................................................................................................... 5
Table 2: Membership Responsibility Table ......................................................................... 5
Final Report: The Charming Rod

Introduction:
Our project has potential entertainment or novelty value for any fan of the Harry Potter series or magic enthusiasts. As the Harry Potter books and movies have become more popular, the idea of creating "magic" can be used to advertise merchandise. With such a product any party host can amaze his or her friends with a simple flick of the wrist. Currently on the market is a universal infrared remote control wand, created by The Wand Company LTD., called the Kymera Magic Wand. The Kymera Magic Wand utilizes simple wand movements to perform up to thirteen commands of any infrared remote control. Our first objective is to essentially recreate the achievements of the already existing device; but design it to be more cost efficient. For future work, we have possible additions that would differentiate our product from the existing one. These additions include, but are not limited to, more complex movements, such as writing digits in the air as input; incorporating voice recognition in coincidence of wand movements for spells; and the addition of lights or a laser pointer to be activated by spells.

Technical Objectives:
The main objective of our project is to design a working motion controlled infrared universal remote control.
- The device casing should look like a wand (which is thin and cylindrically, both of which must be accounted for in design considerations) and be easily wielded by the user (thus, its weight should be minimal).
- The MSP430 micro-controller was chosen because of its low voltage supply requirement of approximately 3.3V. Battery power is required for this handheld device. All device components within the device should be able to be powered by common batteries.
- The continuous operation time of our remote should be comparable to that of any existing battery powered, universal, infrared remote control. The implementation of an on and off switch will help to extend the battery life of the device.

Project Architecture:
The system configuration is displayed in the block diagram that follows.

The MSP430 was chosen for the microcontroller for its low power settings. The battery life of the manufactured product should be comparable to the television remote control. The wand will go into the low power mode when the switch it turned off.

On startup, the wand will act as a universal remote control. The user will point the IR remote control towards the back of the wand and input up to 15 command buttons. The IR receiver will capture each of the commands and store them individually in the memory of the microcontroller.

After the remote control commands are...
Final Report: The Charming Rod

stored in memory the accelerometer begins to send information to the microcontroller. The accelerometer chosen for the wand is the MMA7331LC. This is a 3-axis, analog, 3g accelerometer. The accelerometer will send out a continuous voltage to the 10 bit analog to digital converter (ADC) of the MSP430. The ADC will sample the wave at a specific time interval and store the values in temporary registers. The sequence of channels conversion mode is chosen, such that all 3 axis of the accelerometer can be stored in different temporary registers. The temporary registers are then analyzed to determine the motion of the wand.

When the motion is determined the MSP will send out the corresponding remote control command already stored in memory. An IR transmitter (LED) is used to send out the command. On/Off keying is used to send out the correct IR symbol. A 40Hz square wave corresponds to the value 1 and no wave corresponds to the value of 0.

Manufactured Product design
The components chosen for this product are small enough to fit on a PCB to meet the size objective. The prototype of the manufacture product is pictures below. A cylindrical wooden rod is used as the wand's smaller tip. The IR LED is placed on the tip of the wand. A cardboard casing is used as the handle of the wand to enclose the populated PCB. The switch is accessible on the top of this encasing as well as the indicator LEDs. The battery pack fits compactly with the PCB in the handle is accessible on the bottom to switch out the batteries if needed. In the final manufacturing of the board, the casing will be custom made to enclose all components. This plastic casing will be sturdy and protective for the final product as well as light and manageable to users.

Cost Objectives:
The price of the wand currently on the market is close to $50. Our objective is to manufacture a superior product that has a significant cost reduction to the currently marketed device. The table below shows the bill of material for the final product. As noted the cost of a single wand is $22.48. This table was approximated based on the purchase of single items. This single wand was significantly less than the product currently on the market. When these items are purchased for bulk manufacturing, the cost to produce will be significantly less. For example, when ordered in bulk, the price of the MSP are $2.92.
Therefore the cost objective has been met. The wand can be made at a lesser price than the product currently on the market.

<table>
<thead>
<tr>
<th>Product</th>
<th>Cost (Single)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP430F2272</td>
<td>$6.60</td>
</tr>
<tr>
<td>MMA7331LC</td>
<td>$1.94</td>
</tr>
<tr>
<td>IR Detector</td>
<td>$1.45</td>
</tr>
<tr>
<td>IR LED</td>
<td>$0.49</td>
</tr>
<tr>
<td>Switch</td>
<td>$1.50</td>
</tr>
<tr>
<td>Casing Materials</td>
<td>$10.00</td>
</tr>
<tr>
<td>Resistors &amp; Capacitors</td>
<td>$0.50</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td><strong>$22.48</strong></td>
</tr>
</tbody>
</table>

Table 1: Bill of Materials

### Membership Responsibility Table

<table>
<thead>
<tr>
<th></th>
<th>Wesley Hickle</th>
<th>Jenna Stallings</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR building/testing</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Coding for IR signals</td>
<td>75%</td>
<td>25%</td>
</tr>
<tr>
<td>Alium PCB Design</td>
<td>25%</td>
<td>75%</td>
</tr>
<tr>
<td>Accelerometer testing/coding</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Board Construction</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>50%</td>
<td>50%</td>
</tr>
</tbody>
</table>

Table 2: Membership Responsibility Table

### References:
- [http://www.thewandcompany.com](http://www.thewandcompany.com)
- [http://www.digikey.com](http://www.digikey.com)
- Component datasheets
Final Report: The Charming Rod

Appendices

Appendix A: Altium Schematic

Appendix B: Altium PCB

Appendix C: Code

#include <msp430x22x2.h>

unsigned int i, count, j, k;
unsigned int counter[67], counter2[67], counter3[67], counter4[67];
char temp;
char samp[67], samp2[67], samp3[67], samp4[67];
float temp1, temp2, temp3, temp4, temp5, temp6, temp7;
int main( void )
{
    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;
    DCOCTL = CALDCO_16MHZ;
    BCSC1 = CALBC1_16MHZ;

    P3DIR = 0x31;
    P3OUT = 0x00;
    i = 0;

    for (;;) {
        temp = P3IN & 0x04;
        if (temp == 0x00) {
            P3OUT = 0x10;
            for (;;) {
                temp = P3IN & 0x02;
                if (temp == 0x00) {
                    P3OUT = 0x00;
                    while (i < 67) {
                        if (temp == 0x00) {
                            count = 0;
                            samp[i] = 0x01;
                            while (temp == 0x00) {
                                count = count + 1;
                                temp = P3IN & 0x02;
                            }
                            counter[i] = count;
                            i++;
                        }
                        else {
                            count = 0;
                            samp[i] = 0x00;
                            while (temp == 0x02) {
                                count = count + 1;
                                temp = P3IN & 0x02;
                            }
                            counter[i] = count;
                            i++;
                        }
                    }
                    for (i = 0; i < 60000; i++) {}
                    for (i = 0; i < 60000; i++) {}
                    for (i = 0; i < 60000; i++) {}
                    for (i = 0; i < 60000; i++) {}
                    for (i = 0; i < 60000; i++) {}
                    i = 0;
                    for (;;) {
                        P3OUT = 0x10;
                        temp = P3IN & 0x02;
                        if (temp == 0x00) {
                            P3OUT = 0x00;
                            while (i < 67) {
                                if (temp == 0x00) {
                                    count = 0;
                                    samp2[i] = 0x01;
                                    while (temp == 0x00) {
                                        count = count + 1;
                                        temp = P3IN & 0x02;
                                    }
                                    counter2[i] = count;
                                    i++;
                                }
                                else {
                                    count = 0;
                                    samp2[i] = 0x00;
                                    while (temp == 0x02) {
                                        count = count + 1;
                                        temp = P3IN & 0x02;
                                    }
                                    counter2[i] = count;
                                    i++;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
Final Report: The Charming Rod

count = 0;
samp2[i] = 0x00;
while (temp == 0x02) {
    count = count + 1;
    temp = P3IN & 0x02;
}    
counter2[i] = count;    
i++;    
}

for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
i = 0;
for (;;) {
P3OUT = 0x10;
temp = P3IN & 0x02;
if (temp == 0x00) {
P3OUT = 0x00;
while (i < 67) {
    if (temp == 0x00) {
        count = 0;
        samp3[i] = 0x01;
        while (temp == 0x00) {
            count = count + 1;
            temp = P3IN & 0x02;
        }    
counter3[i] = count;
    i++;
}    
else {
    count = 0;
    samp3[i] = 0x00;
    while (temp == 0x02) {
        count = count + 1;
        temp = P3IN & 0x02;
    }    
counter3[i] = count;
    i++;
}    
}    
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
i = 0;
for (;;) {
P3OUT = 0x10;
temp = P3IN & 0x02;
if (temp == 0x00) {
P3OUT = 0x00;
while (i < 67) {
    if (temp == 0x00) {
        count = 0;
        samp4[i] = 0x01;
        while (temp == 0x00) {
            count = count + 1;
            temp = P3IN & 0x02;
        }    
counter4[i] = count;
    }    
}    
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
in++;
for (;;) {
P3OUT = 0x10;
temp = P3IN & 0x02;
if (temp == 0x00) {
P3OUT = 0x00;
while (i < 67) {
    if (temp == 0x00) {
        count = 0;
        samp5[i] = 0x01;
        while (temp == 0x00) {
            count = count + 1;
            temp = P3IN & 0x02;
        }    
counter5[i] = count;
    }    
}    
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
for (i = 0; i < 60000; i++) {}
i++;
}
else {
    count = 0;
samp4[i] = 0x00;
    while (temp == 0x02) {
        count = count + 1;
temp = P3IN & 0x02;
    }
    counter4[i] = count;
i++;
}
}

ADC10CTL0 &= 0x0000; //disable ADC10 conversions
ADC10CTL0 |= 0x00090; //set sample-and-hold time = 16 x 16C10CLKS,
               //set to automatic conversions,
               //turn ADC10 on
ADC10AE0 = 0x07; //enable pin A3 for analog input
ADC10CTL1 = 0x2002;
ADC10CTL0 |= 0x0003;
temp4 = ADC10MEM & 0x03FF;
temp5 = ADC10MEM & 0x03FF;
temp6 = ADC10MEM & 0x03FF;
for (;;) {
P3OUT = 0x20;
temp1 = ADC10MEM & 0x03FF;
temp2 = ADC10MEM & 0x03FF;
temp3 = ADC10MEM & 0x03FF;
if (temp3 >= 0x1FF && temp1 < 0x307 && temp2 < 0x0307) {
P3OUT = 0x00;
for (i = 0; i < 67; i++) {
    if (samp3[i] == (0x01)) {
        for (j = 0; j < counter3[i]/24; j++) {
P3OUT = 0x01;
        for (k = 0; k < 11; k++) {}
P3OUT = 0x00;
        for (k = 0; k < 11; k++) {}
    }
}
if (samp3[i] == (0x00)) {
    for (j = 0; j < counter3[i]/24; j++) {
P3OUT = 0x00;
    for (k = 0; k < 11; k++) {}
P3OUT = 0x00;
    for (k = 0; k < 11; k++) {}
}
}
if (temp1 >= 0x307 && temp4 <= 0x304 && temp2 < 0x0307 && temp3 < 0x0303) { //temp1 from 253 to 309 305 305
P3OUT = 0x00;
for (i = 0; i < 67; i++) {
    if (samp4[i] == (0x01)) {
        for (j = 0; j < counter4[i]/24; j++) {
P3OUT = 0x01;
        for (k = 0; k < 11; k++) {}
P3OUT = 0x00;
        for (k = 0; k < 11; k++) {}
    }
}
}
}
}
if (samp[i] == 0x00) {
    for (j = 0; j < counter[i]/24; j++) {
        P3OUT = 0x00;
        for (k = 0; k < 11; k++) {}
        P3OUT = 0x00;
        for (k = 0; k < 11; k++) {}
    }
}

if (temp2 > 0x308 && temp1 < 0x0310 && temp3 < 0x0310) { //309 305 305
    P3OUT = 0x00;
    for (i = 0; i < 67; i++) {
        if (samp2[i] == 0x01) {
            for (j = 0; j < counter2[i]/24; j++) {
                P3OUT = 0x01;
                for (k = 0; k < 11; k++) {}
                P3OUT = 0x00;
                for (k = 0; k < 11; k++) {}
            }
        }
        else {
            for (j = 0; j < counter2[i]/24; j++) {
                P3OUT = 0x00;
                for (k = 0; k < 11; k++) {}
                P3OUT = 0x00;
                for (k = 0; k < 11; k++) {}
            }
        }
    }
}

    temp4 = temp1;
    temp5 = temp2;
    temp6 = temp3;
    ADC10CTL0 |= 0x0001;
}
}