

```

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <string.h>
#include <SimpleTimer.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars
and 2 line display
int c = 6000; // for timing
//////////Daniel's code//////////
int DEF_STATE;
int lazerPin = 8; // lazer output
SimpleTimer timerSend;
int tsID; // timer ID
int st = 0; // state of lazer
int bt = 0; // pulse width
int btLow = 50; // short pulse width
int btHigh = 250; // long pulse width
//////////
void setup() {
    Serial.begin(9600); // Open serial monitor at 115200 baud to see ping results.
    lcd.init(); // initialize the lcd
    lcd.backlight(); // turn on backlight
    lcd.clear(); // clear the display
    for (int i = 2; i < 7; i++) {
        pinMode(i, INPUT_PULLUP);
    }
    lcd.print("Ready"); // Print line 1 message to the LCD
    DEF_STATE = digitalRead(2);

    //////////Daniel's code//////////
    pinMode(lazerPin, OUTPUT);
    //////////
}

int count = 0;
int trig;
int station = 0;
String msg = "";

void loop() {
    for (int i = 2; i < 7; i++) { // read each button state
        int state = digitalRead(i);
        if (state != DEF_STATE) {
            runcmd(i);
        }
    }
    if (count == 3000) { // basically a timer
        lcd.clear();
    }
}

```

```

    lcd.print("Ready");
    count++;
} else if (count < 3000) {
    count ++;
}
//////////Daniel's code//////////
timerSend.run();
if (Serial.available() > 0 && st == 0) { // signal from computer
    int inByte = Serial.read();
    switch (inByte) {
        case 'l': //lamp
            bt = btLow;
            beginSend();
            break;
        case 't': //tea
            bt = btHigh;
            beginSend();
            break;
        case 's': // change station
            runcmd(2);
    }
}
//////////
}

void runcmd(int inpt) { // run command of button pressed
    char cmd = ' ';
    switch (inpt) {
        case 2:
            station++;
            if (station > 8) {
                station = 0;
            }
            Serial.write('s');
            Serial.write(station);
            cmd = '\r\n';
            msg = "Station changed";
            count = -3000;
            break;
        case 3:
            cmd = ')';
            msg = "Volume increased";
            count = -3000;
            break;
        case 4:
            cmd = '(';
            msg = "Volume decreased";
            count = -3000;
    }
}

```

```
        break;
    case 5:
        cmd = 'p';
        msg = "Play/Pause";
        count = -3000;
        break;
    case 6:
        cmd = 'n';
        msg = "Song Skipped";
        count = -3000;
        break;
    }
    Serial.write(cmd);
    lcd.clear();
    lcd.print(msg);
    delay(200);
}

//////////Daniel's code//////////
void beginSend() { // start lazer
    digitalWrite(lazerPin, true);
    tsID = timerSend.setTimer(bt, stopSignal, 1); // stop after pulse width
    st = 1;
}
void stopSignal() { // stop lazer
    digitalWrite(lazerPin, false);
    st = 0;
}
//////////
```