

## Final Report: Load Sensor Alarm Clock

### I. Introduction

#### A. Problem Statement:

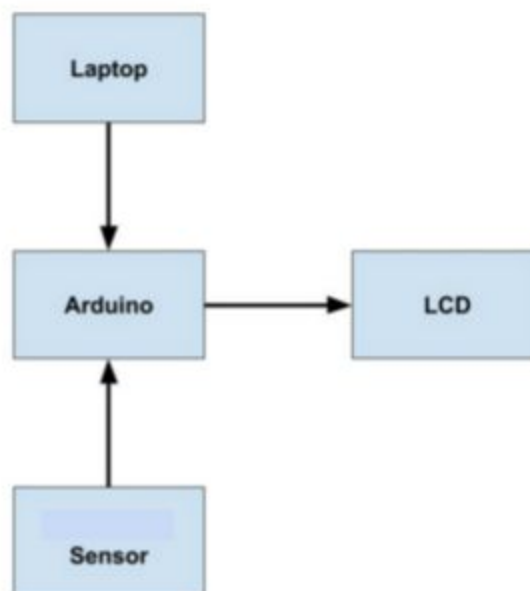
- a. Traditional alarm clocks have a snooze feature that permits a user to silence the alarm and remain in bed instead of getting up. For users that have trouble getting out of bed in the morning, such as the average college student, such a system has the potential to allow for oversleeping and tardiness to events.

#### B. Solution Overview:

- a. Requiring the user to rise from the bed in order to silence the alarm clock increases the effectiveness of the alarm by forcing the user into motion. A load sensor placed below a user's mattress and linked to the alarm clock allows for detection of the user's presence in the bed. While the load sensor registers an added weight to the mattress equivalent to a human body, the alarm continues to make noise once triggered. Until the user rises from the bed and the load sensor no longer detects an added weight, the alarm cannot turn off.

### II. Design

#### A. Block Diagram

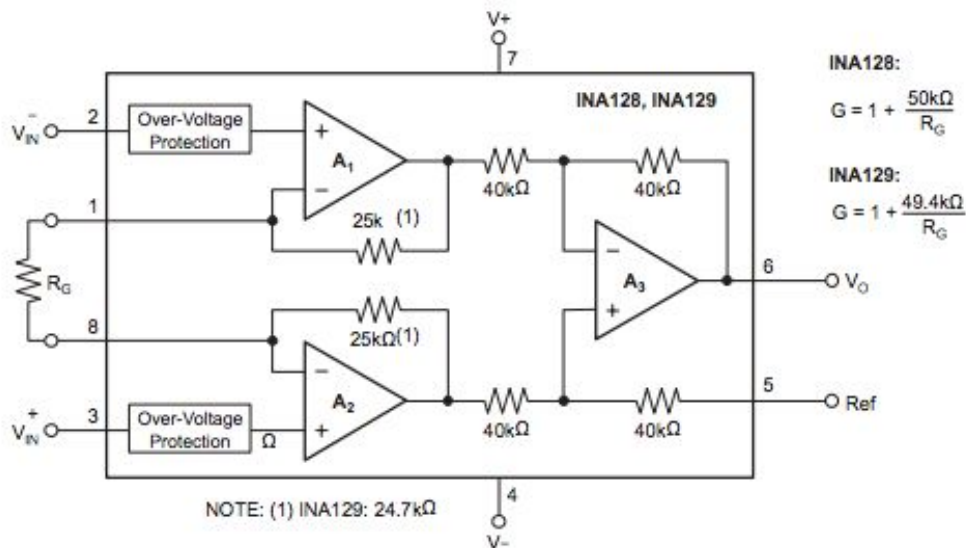


[Figure 1: Load Sensor Alarm Clock Block Diagram]

## B. Written Descriptions of Block Diagram

- a. The laptop acts as a power source and also uploads the codes to the arduino to perform its task. The sensor is also connected to the arduino to give the arduino the sensor's values. These values will tell the arduino when to activate its specific command, as determined by the codes from the laptop (the values are also read on the laptop). When the arduino's function is triggered by the sensor and laptop, it activates the LCD to where it can perform its alarm function.

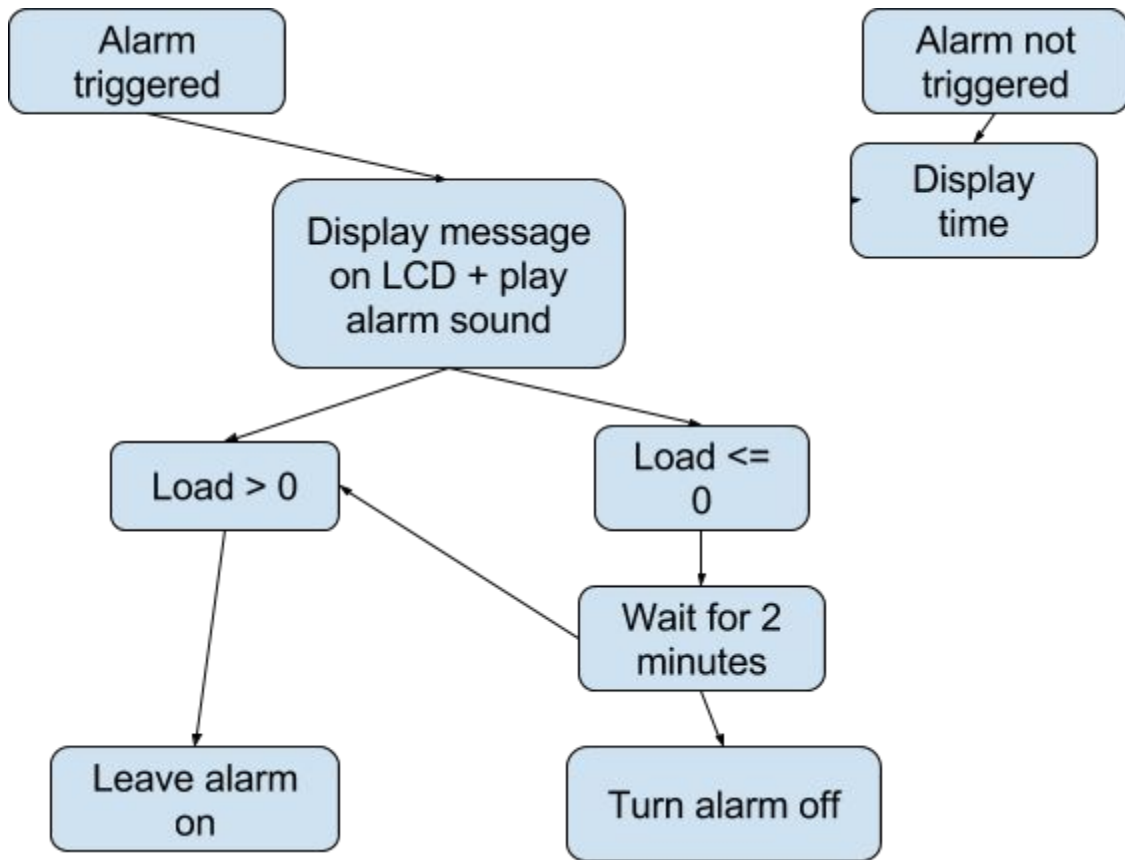
## C. Circuit Diagram



[Figure 2: Instrumentation Amplifier Schematic]

In order to read values from the pressure sensor, it was necessary to build an instrumentation amplifier circuit in order to adjust the signal from the pressure sensor to a value readable by the Arduino. The pressure sensor came in the form of a strain gauge, which provides a measurement based on the fact that the metal of the gauge flexes when pressure is applied to it. This changes the area of the metal, which in turn slightly alters the resistance of the sensor. This minute change in resistance can be converted to an accurate weight measurement, but requires amplification in order to register on the Arduino. The circuit in Figure 2 replicates an instrumentation amplifier from the Texas Instruments INA12x series of parts, which utilizes three operational amplifiers and resistors in order to amplify a signal. The most important part of the circuit is the gain resistor ( $R_G$ ) which greatly affects the overall amplification of the signal. Altering the value of the gain resistor can amplify the signal to anywhere from 10 to 1000 times the original value.

D. Flowchart of Software



E. Code Used

```

1 // include the library code:
2 #include <LiquidCrystal.h>
3 #include <Time.h>
4 #include "pitches.h"
5
6 //setup time characteristics
7 #define TIME_HEADER "T" // Header tag for serial time sync message
8 #define TIME_REQUEST 7 // ASCII bell character requests a time sync message
9
10 int alarmh = 7;
11 int alarmm = 30;
12 int alarms = 0;
13
14 time_t t = now();
15
16 int h = 7;
17 int m = 29;
18 int s = 54;
19
20 // initialize the library with the numbers of the interface pins
21 LiquidCrystal lcd(9, 8, 5, 4, 3, 2);
22
23 //set up sound characteristics
24
25 // notes in the melody:
26 int melody[] = {
27   NOTE_C4, NOTE_G3, NOTE_C4, NOTE_G3, NOTE_C4, NOTE_G3, NOTE_C4, NOTE_G3
28 };
29
30 // note durations: 4 = quarter note, 8 = eighth note, etc.:
31 int noteDurations[] = {
32   4, 8, 4, 8, 4, 8, 4, 8
33 };
34
35 //setup load characteristics
36 int load = 0;
37
38 //alarm on/off indicator
39 int alarmOn = 0;
40
41 //delay for user to remain out of bed
42 int check = 100000000;
43
44 void setup() {
45   // set up the LCD's number of columns and rows:
46   lcd.begin(16, 2);
47 }
48
49
50
51 void loop() {
52   if(alarmOn == 0){
53     //read load
54     load = analogRead(0);
55
56     // set the cursor to column 0, line 1
57     // (note: line 1 is the second row, since counting begins with 0):
58     lcd.setCursor(0, 1);
59     delay(1000);
60     s = s+1;

```

```

63
64  if(s == 60){
65      m++;
66      s = 0;
67  }
68
69  if(m ==60){
70      h++;
71      m = 0;
72  }
73
74  if(h==12){
75      h = 0;
76  }
77
78  lcd.clear();
79  lcd.print(h);
80  lcd.print(m);
81  lcd.print(s);
82
83  if(h == alarmh && m == alarmm && s ==alarms){
84      alarmOn++;
85  }
86
87  }
88  while(alarmOn == 1){
89
90      //read load
91      load = analogRead(0);
92
93      if(load > 0){
94
95          //read load
96          load = analogRead(0);
97
98          lcd.clear();
99          lcd.setCursor(0,0);
100         lcd.print("Wake up!");
101
102         // iterate over the notes of a melody:
103         for (int thisNote = 0; thisNote < 8; thisNote++) {
104
105             int noteDuration = 1000 / noteDurations[thisNote];
106             tone(10, melody[thisNote], noteDuration);
107             int pauseBetweenNotes = noteDuration * 1.30;
108             delay(pauseBetweenNotes);
109             // stop the tone playing:
110             noTone(10);
111
112             //prevent user from returning to bed for a certain amount of time
113         }else{
114             delay(check);
115             //read load
116             load = analogRead(0);
117             if(load <=0){
118                 alarmOn = 0;
119             }
120         }
121     }
122 }
123 }
124 }

```

### III. Results

#### A. Present Results

- a. For our results, we were unable to produce a functioning alarm clock from our sensor. This can be attributed to the fact that our implementation of the instrumentation amplifier did not work as expected. Therefore, the project had the same basic functionality as a regular alarm clock but did not extend to the most important aspect, which was the pressure sensing portion of the plan.

#### B. Qualitative Analysis of results

- a. Our pressure sensor did not function the way it should have. We did not receive valid values from the sensor, which in turn did not allow the alarm to receive the function of when to stop beeping. The alarm simply went off when triggered and did not stop based on force applied to the sensor. However, the melody from the speakers and the display on the LCD functioned properly, with an appropriate sound and time and message display observed during operation.

#### C. Quantitative Analysis of results

- a. The Arduino was unable to read significant values from the pressure sensor in order to complete that portion of the clock. Although we applied force to the strain gauge after appropriately connecting it to the instrumentation amplifier and the Arduino, the change in values read by the Arduino essentially amounted to 0. We attempted to diagnose the problem by hooking the instrumentation amplifier to an oscilloscope and the pressure sensor to a multimeter. The pressure sensor displayed minute changes in voltage when monitored by the multimeter (ie. minute changes in the resistance of the sensor) amounting to about 0.001 V on average. However, when we monitored the instrumentation amplifier with the oscilloscope and applied 500 mV across the circuit, the resulting voltage still read as 500 mV, indicating that no amplification had occurred across the ends of the circuit. Furthermore, the visual signal had a large amount of noise.

### IV. Future Work

#### A. Next Steps for Project

- a. Further debugging of the instrumentation amplifier circuit is required. Theoretically, altering the value of the gain resistor should generate a less noisy reading. Another useful addition would be an Arduino time chip in

order to generate real time readings on the alarm clock instead of the hardcoded version we currently have.

## V. Conclusion

### A. What Worked?

- a. The alarm clock portion of the project worked as anticipated. We were successfully able to set up a speaker and LCD in conjunction with the Arduino such that it displayed the time appropriately until the alarm was triggered. At this point, the screen displayed a “Wake up!” message and the speaker generated a noise. Although the alarm clock did not display real time data due to the absence of a time chip, the overall setup worked about as well as could be expected.

### B. What Didn't Work?

- a. Due to time constraints, the instrumentation amplifier portion of the circuitry did not function as expected. Rather than amplifying the signal from the pressure sensor, the circuit appeared to either have no effect or perform in opposition to its objective by reducing the signal. The signal further appeared to contain a lot of noise when monitored by an oscilloscope.

### C. What Did You Learn?

- a. Although the clock did not really function properly, we learned a lot about the functionality of a strain gauge through experimenting with it. For example, we learned that a strain gauge provides a measurement due to the fact that the metal of the gauge flexes when pressure is applied to it. Since this changes the area of the metal, a slight alteration in the resistance of the metal occurs which, when amplified and appropriately converted, can provide an accurate weight measure. We also learned a good deal about operational and instrumentation amplifiers. These were not topics covered in ECE 110, so they were confusing at first, but some research soon cleared up some of our questions. Many practical uses of such a circuit could be imagined beyond just amplifying the reading from a strain gauge, so this was probably the most useful aspect of our project we learned about.