



Institutt for informatikk
Universitetet i Oslo

Technical Report

IN1060 - Bruksorientert design

TeamUino

Siri Sollerud

Stela Ceaicovscaia

Zach Erjohn Kobe Gonsholt

Yan Jiang



Introduction

The topic of our group is to discover new ways of reducing food waste in grocery stores using IT solutions. We chose to focus on grocery store employees and aim to provide a digital solution to help to smooth their work process in the 'too good to go' service and also advocate 'too good to go' App to raise people's awareness of food waste.

After several cycles of user-centered design and iterations, we present a self-service box using arduino which can reduce customers' waiting time, reduce workload for employees and advocate the App to a bigger audience. The customer can enter code using a number pad and receive feedback on screen. When the password is correct and accepted, the motor will rotate and unlock. The box's appearance is also elaborately designed and decorated to promote the 'too good to go' App.

Video

Our video first shows a scenario that shows the problem to the employees and customers in a Coop Mega grocery store. Where an employee was already tired after a hectic day and had to fetch 'magic bag' to the customers again and again. The customers were also not satisfied with the long waiting time. Then we show a short animation video on how our self-service lockbox works. The video then shows how our prototype can be used in the store. How the employee can put bags in the box and customers can come and unlock the box to get their bag without waiting. Our product (prototype) still can not perform stably in the grocery store environment with real users due to some connection issues, therefore we have adopted a Wizard of Oz approach in some parts of the prototype and video. All the people in our video agreed to being filmed in advance.

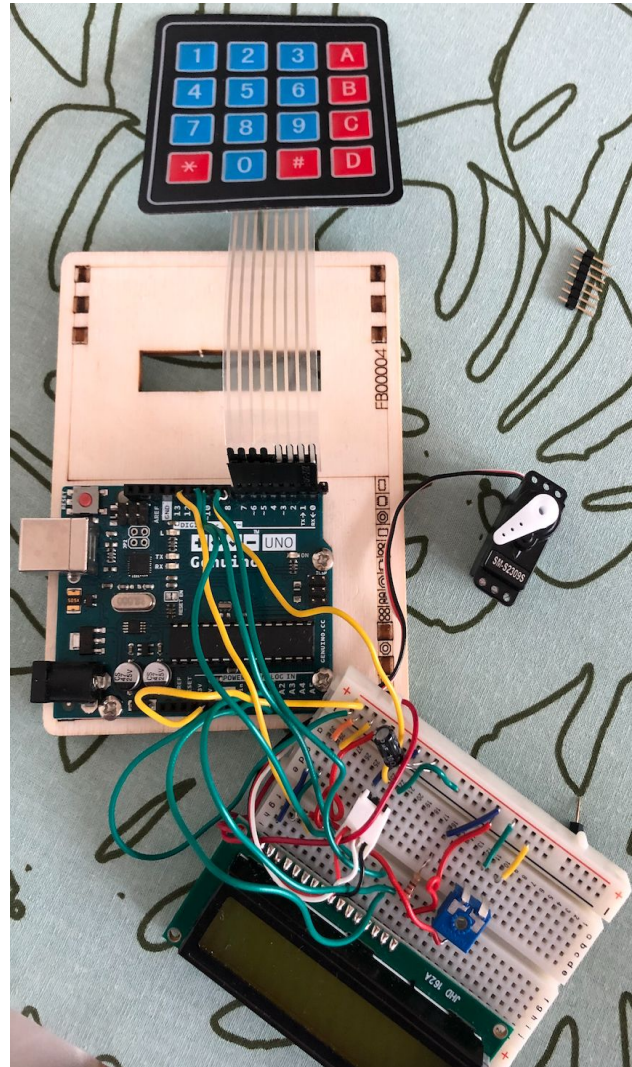
Video Link:

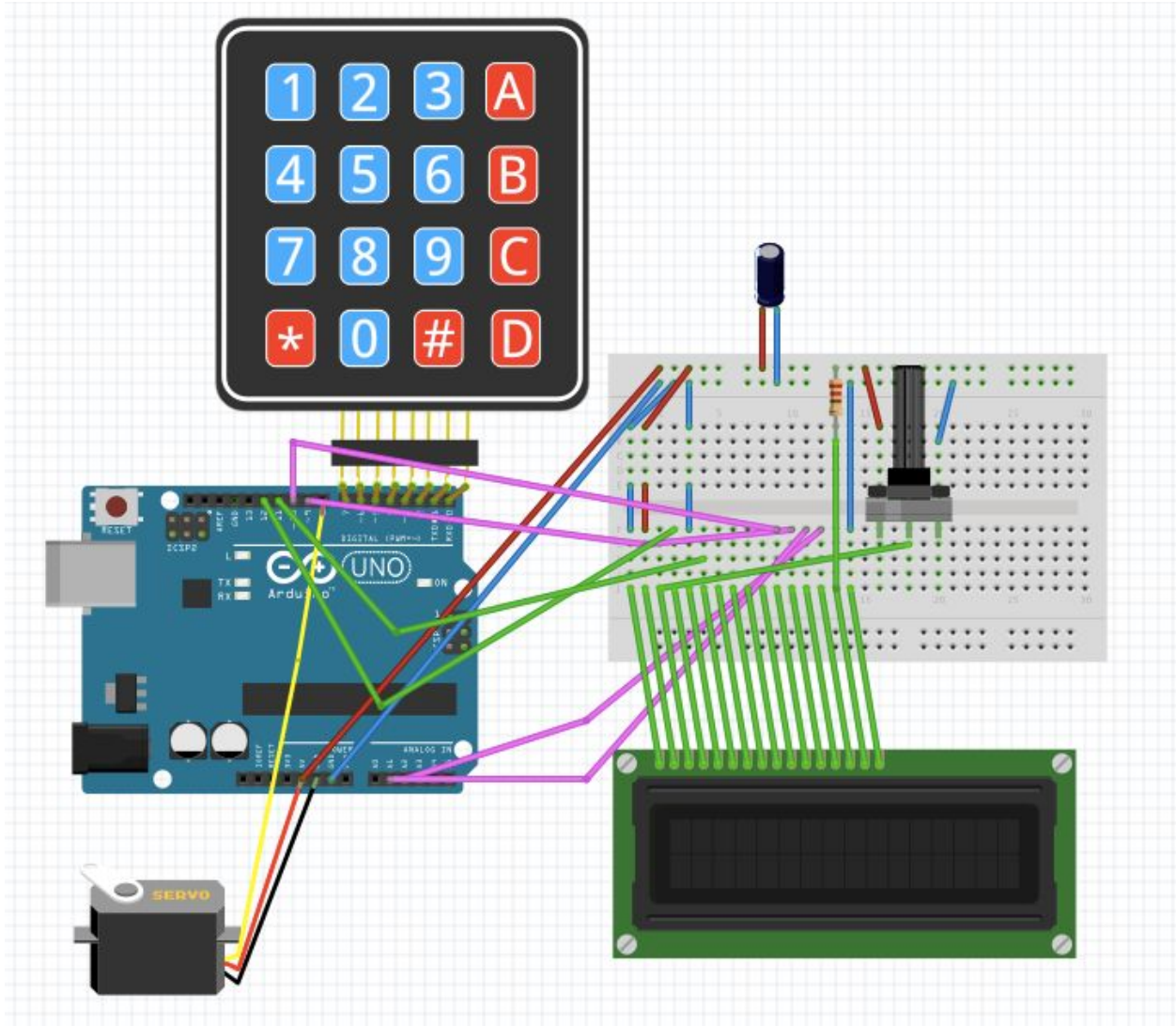
<https://youtu.be/LmZxXhG1uP8>

Technical solution

As early as our first iteration, we began to think practically which Arduino parts can be used to realize the functions that we want in our prototypes. In our third iteration, we began to work on arduino. We used switched buttons as user input and LCD screen as user feedback. We also compare the servo motor and dc motor and settle with servo motor as it is more easily controlled in our case. When the servo motor starts to rotate, it draws more current than if it were already in motion, so we also have to place a 100 uf capacitor across power and ground.

Later, we receive feedback from the users that the switch is very difficult to use and with very limited password selection as there were only two switches connected to the pins. Therefore, in the fourth iteration, we ordered 4*4 keypad and replace the switches so that it appears more like a system that we use in real life. Now we can have input from 0 to 9 and letters A-D as shown in the figure.





The system begins with greeting the users welcome to 'too good to go' as shown in our code. After the greeting, the system waits for user input. When the user enters passwords, we compare it with the password that we set. If it is correct, the screen will show that it is correct and the box is not unlocked. Then, the motor receives signals and rotates 90 degrees. And then our box is unlocked, users can open it and get their magic bag.

The components of this product is showed below:

Components	Numbers
Arduino Uno	1
LCD screen	1
Servo motor	1
4*4 keypad	1
Jumper wires	10+
Capacitor	1
Potentiometer	1
Resistors	1
Breadboard	1
USB cable	1
Male header pin	10+

Code

Codes and explanations see this link or see screenshots below :

https://create.arduino.cc/editor/yan2030_/ff48a5b1-f7b6-44c9-86c3-1f2717f9cc29/preview

```

1 //this is a self-service lockbox for the user of 'too good to go' App in grocery store Coop Mega
2 //it examine the user's input, if the entered code is correct, then the box will be lock/unlock
3
4
5 //setting up the lcd
6 #include <LiquidCrystal.h>
7
8 // initialize the library with the numbers of the interface pins, using some of the analog pins as digital pins
9 LiquidCrystal lcd(12, 11, 10, 9, A1, A2);
10 #include <Servo.h>
11
12 //innitalize the motor
13 Servo myServo;
14
15 //setting up the keypad
16 #define Password_Length 4
17 #include <Keypad.h>
18
19 const byte ROWS = 4;
20 const byte COLS = 4;
21
22 char hexaKeys[ROWS][COLS] = {
23   {'1', '2', '3', 'A'},
24   {'4', '5', '6', 'B'},
25   {'7', '8', '9', 'C'},
26   {'*', '0', '#', 'D'}
27 };
28 //if using pin 0 and 1, then should not use Serial print.
29 byte rowPins[ROWS] = {7, 6, 5, 4};
30 byte colPins[COLS] = {3, 2, 1, 0};
31
32 Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
33

```

```

33
34 //set a password, can be changed
35 char Data[Password_Length];
36 char Master[Password_Length] = "238C";
37 byte data_count = 0, master_count = 0;
38 bool Pass_is_good;
39 char customKey;
40
41
42 void setup() {
43   myServo.attach(8);
44   myServo.write(0);
45   //this is a greeting message that prints on the screen
46   greeting();
47   //clear the screen
48   lcd.clear();
49
50
51 }
52
53 void loop() {
54   //if user input is activated, then add it to the data
55   customKey = customKeypad.getKey();
56   if (customKey){
57     Serial.println(customKey);
58     Data[data_count] = customKey;
59     lcd.setCursor(data_count,1);
60     lcd.print(Data[data_count]);
61     data_count++;
62

```

```

63     }
64 }
65 // when the length of the user input is same as our password, compare to see if the password is correct
66 if(data_count == Password_Length){
67     delay(2000);
68     lcd.clear();
69     // lcd.print(Data);
70     // lcd.setCursor(0,1);
71     //lcd.print(Master);
72     delay(2000);
73
74
75     //if the password is correct, print to the screen and unlock, rotate motor 90 degrees
76 if(!strcmp(Data[0,3], Master[0,3])){
77     lcd.print("Correct");
78     lcd.clear();
79     lcd.print("code accepted");
80     lcd.setCursor(0, 1);
81     if (locked == true){
82         lcd.print("***unlock**");
83         delay(2000);
84         myServo.write(90);
85         delay(2000);
86     }else{
87         lcd.print("***Locked**");
88         delay(2000);
89         myServo.write(90);
90         delay(2000);
91     }
92 }

```

```

93 //if the password is not correct, give user feedback
94 else{
95     lcd.print("Incorrect");
96     data_count=0;
97     delay(1000);
98     greeting();
99 }
100 //clear the data
101 lcd.clear();
102 clearData();
103
104 }
105 }
106 //the method to clear the data, so it wont mess up with new user input
107 void clearData(){
108 while(data_count !=0){
109     Data[data_count--] = 0;
110 }
111 return;
112 }
113
114
115 void greeting() {
116     lcd.begin(16, 2);
117     lcd.print("Too good");
118     lcd.setCursor(0, 1);
119
120     lcd.print("to go :)");
121     delay(3000);
122     lcd.clear();
123 }

```

```
123  
124   lcd.print("Welcome to self-");  
125   lcd.setCursor(0, 1);  
126   lcd.print("service lockbox");  
127   delay(2000);  
128   lcd.clear();  
129  
130 }  
131
```
