대본을 만들어 보자!

have you ever been to a busy bar? Sometimes bar is too busy that we can't even order our drinks, or the employee forgets our order. This is where our idea started. We are going to create a smart bottle opener that has a function of ordering drinks by sensoring the color sticker attached to the table. By our smart bottle opener, we don't have to worry about whether the employee forgets our order.

This is how our smart opener works. [ppt1] as seen in this picture,

남택완

1. 발표지원 시뮬레이션 소프트웨어 개발을 맡았고 스마트오프너를 이해하기쉽게 설명하기위한 시뮬레이트 프로그램을 Visual Basic.net 기반으로 개발함.

Taekwan made presentation-supporting software development. It helps to understand our Smart Opener. It was developed by Visual Basic. net

2. Count버튼을 누르면 한번에 주문할 수량이 1씩 증가함.

See here, when you press the count button, the number of order increases by 1.

3. 컬러센서를 표현하기위해 컬러를 값을 선택할수있는 버튼을 구현하여 해결함.

And remember our color sensor? We made these color button to implement the color sensor since it was impossible to implement the real color sensor.

4. Order 버튼을 누르면 지금까지 누적된 count 값과 술 종류를 리턴함.

And When you press the 'order'button, it returns all the kinds of drink and accumulated drink number.

5. Cancel버튼을 누르면 컬러에 따른 술종류별로 1씩 주문 취소됨.

And when you press the cancel button, it cancels 'one' order in each kind of color.

6. 첫번째 주문일 경우 테이블 현황에 해당 테이블을 빨간색으로 표시하여 테이블이 사용중임을 표시하고 사용가능한 총 테이블수가 1감소됨.

And if it is the first order, it presents current table color from gray to red so we can present the table is occupied, and remaining table number decreases by one.

7. 테이블을 식별할수있는 테이블 코드 7-세그먼트 1개와 주문한 술 종류를 식별할수있는 코드를 표시하는 7-세그먼트 2개 따라서 7-세그먼트 총3개로 이루어진 전광판에 주문사항을 표시함.

Electronic display displays table code by 7-segment, and two 7-segments that can present kind of drink that we ordered. So totally, we have three 7-segment electronic display to show our orders.

8. 주문 테이블과 수량, 술종류를 저장함.

And we will save ordering table number, number of drinks, kind of drinks we ordered.

9. 실시간으로 영수증 인터페이스에 반영하여 누적된 주문 정보 업데이트함.

And it will instantly update accumulated ordering information on the receipt interface.

10. 리셋버튼을 누르면 해당 테이블의 누적 정보를 삭제하고 테이블 현황에 해당 테이블을 녹색으로 표시하여 테이블이 사용가능함을 표시하고 사용가능한 총 테이블수가 1증가됨.

정현규

Id2 7 segment decoder

I will briefly explain about 7 segment decoder being used in our project.

Since our CAU pub has 7 different types of beverages being sold to consumers, we thought that we could consider them as inputs, expressing all those types in 3 bits and display the names of each drink on the 7 segment LED. Because we cannot show enough information about the names of drink when displaying only one 7 segment, we used a couple of 7 segments with the alphabets ranging from a to n, the parts of those two 7 segments. We then chose two alphabets from the actual names of each beverage in order to display them on the 7 segments for example, cl from cola sj from soju. By doing so, we can tell the staff what types of drink the consumers have ordered.

C2, C1, C0 are the 3 bits input codes. When input codes go through the decoder, it will display the corresponding names of drink on the 7 segments. To do so, I filled the 7 segment decoder truth table, just like what we did in the midterm exam, the 노가다 thing. I then drew k-maps of each of those 7 segments parts and simplified all the equations from the k-maps that you can see down there.

Using Quartus and all that information, I made the 7 segment decoder that can display the names of each drink.

김건우

Add의 역할은 주문한 술의 종류별 술병의 개수를 전체 술의 가격으로 나타내주는 것입니다. 카운터로부터 얻은 주문한 술의 종류별 술의 개수를 input으로 받아서, 종류에 따른 gravity를 쉬프트 처리하여 모두 더한 후 전체 gravity를 구하고 전체 술의 가격인 4(,000)을 부가하기 위해 쉬프트를 2번 해줍니다. 술의 종류는 7개이므로 6개의 Adder가 필요하고 최대 주문량을 고려하여 넉넉히 8비트 Adder를 만들었습니다.

Function of Adder is to get sum of accumulated expense. The input is number of order of drinks. Since drinks have gravity, we bit-shifted numbers and added the whole. And, to get the price 4,000(4 thousand won), we shifted the whole price twice. There are 7 kinds of drink so we need Six adders, and considering that .///보충필요

오혜성

1. opener

plus_BTN 또는 minus_BTN을 누르게 되면 beverage_id를 인식한다.

when you press plus button or minus button, they recognize beverage id.

만약 plus_BTN을 누르면 order값이 증가하고 minus_BTN을 누르면 order값이 감소한다.

And if plus button is pressed, value of order increases and if minus button is pressed, value of order decreases.

그리고 order값은 opener_order_output을 통해서 값이 출력되며, display에 현재 order값이 표시된다.

And 7-segment display will show the present value of order.

그리고 send_BTN을 누르면 레지스터로 table_num, beverage_id, order_num 값이 전송되며, table_num, beverage_id, order_num값은 초기화 된다.*/

and when you press the send_button, the data of [table number, beverage id, number of drinks] are sended to register, and [table number, beverage id, and order number] will be initiallized to zero.

Bill Calculator

Our Color sensor can distinguish seven colors so we have total seven Beverage_ID. When calculating the total price, we have to include number of drinks that have been ordered for specific Beverage ID and that Beverage ID's weight.

To calculate the total price easy, we made concept called weight. One weight equal to 4000 won, so two weight means 8000won. Therefore, to calculate total price we can use below pseudo code.

```
while(Beverage_ID)
  price += num * weight * 4000;
```

In order to do this calculation, we use two 4-bit full adder to make 8-bit full adder. After that, we get the value of Beverage_ID and num output from opener remote control. Beverage_ID's value is between 000 and 111 including 111. Because these values have to be all separated and have to go to Adder as an input, before it goes in to adder's input from D-FlipFlop, in order to make the value num to be separated by Beverage_ID from D-FlipFlop, when and operation of Beverage_ID's 0bit, 1bit, 2bit equals 1, value num enters to Adder as an input.

Also, when the weight is one, we make Beverage_ID's 0bit to go in to 0bit's place, 1bit to go in to 1bit's place. When weight is two it means twice a value so we bit shift once. So, when the weight is two, we make Beverage_ID's 0bit to go in to 1bit's place 1bit to go in to 2bit's place.

Finally, When calculating the final price we can just multiply 4 so we do bit shift twice(two bit shift) and send that values to Adder as an input.

조예진

random

random function is made to randomly pay bill with friends. It is only for fun, so instead of real number generator, we used pseudo random number generator. We have 4-people table so we have to make random number from 0 to 3.
This table shows how the circuit I showed makes number from 0 to 31. we need 2 bit, so it makes 31 multiplied by 31, equals 961. I only used x1 part and parallaly attacked two.

if random number is returned 01, then the number 1 person pays the whole price.