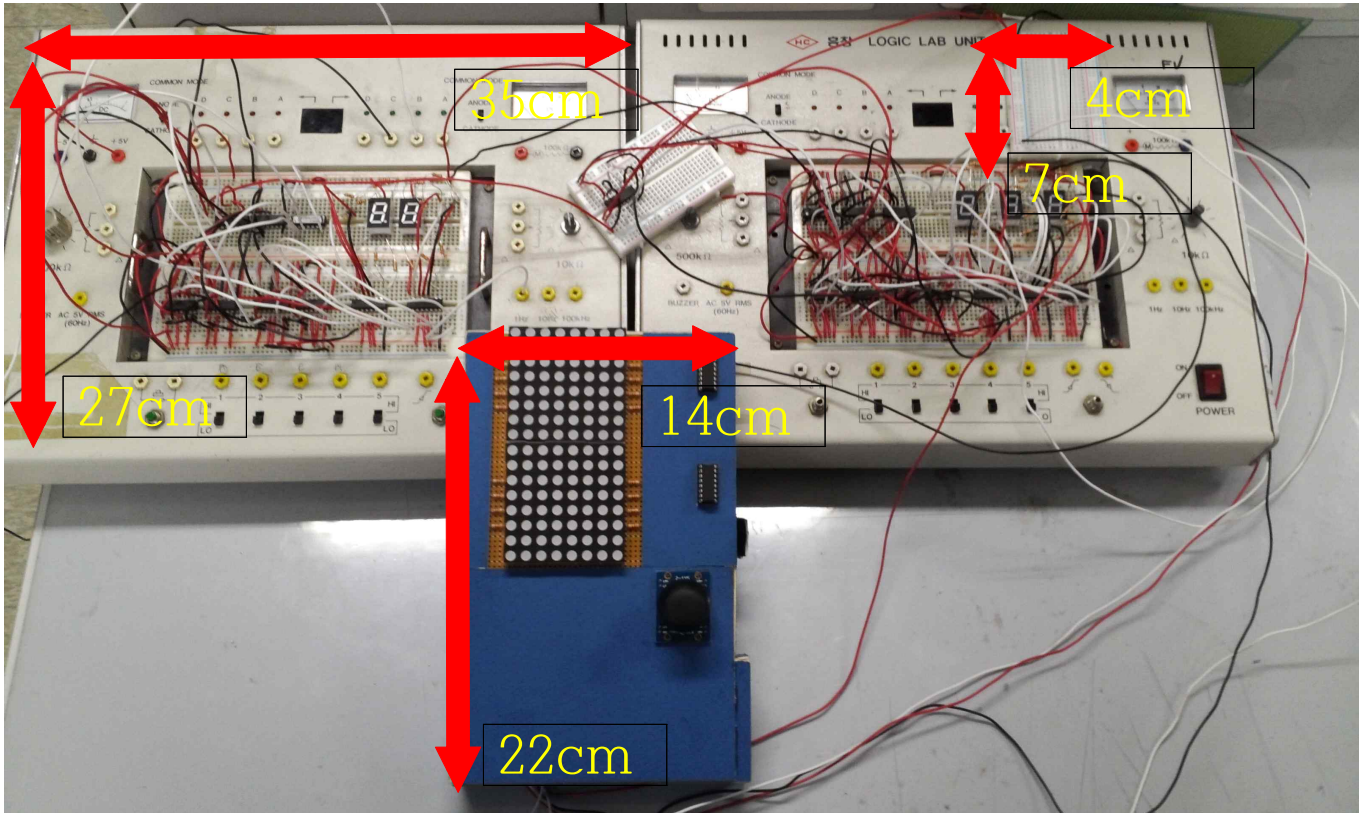



## [ 목 차 ]

1. 요구규격
2. 설계제한 요소
3. 설계방법 및 동작 원리
4. 설계도
5. 부품
6. 구현 및 실험/평가
7. 프로젝트 소감

## 1. 요구규격



[그림 1] 작품 규격

구분	시스템 요소명	상세 요구 규격
H/W	IC	<ul style="list-style-type: none"> <li>▷ NOT Gate</li> <li>▷ Shift register</li> <li>▷ BCD Counter</li> <li>▷ BCD to 7-segment</li> <li>▷ 4bit full adder</li> <li>▷ 4bit latch</li> </ul>
	LED Matrix, 7-segment	<ul style="list-style-type: none"> <li>▷ 시각적 출력을 위한 8x8(*2) LED Matrix와 7-segment</li> <li>▷ LED Matrix : Arduino에서 생성된 신호를 Shift Register의 신호 분배에 따라 (<math>x_n, y_n</math>)의 위치에 LED가 점멸함.</li> <li>▷ 7-segment : 한 자릿수 당 0-9 까지의 숫자를 시각적으로 출력하며, 각 자릿수에 CLR값의 지연시간을 다르게 주어서 신호의 오류를 방지함.</li> </ul>
	Bread board	▷ 회로 구성 및 IC 장착 과 연동
	2-Axis Joystic	▷ 테트리스 블록의 이동(상, 하, 좌, 우)신호 값을 생성 하여 Arduino에 신호를 보내어 블록의 이동 및 회전(90°)을 제어한다.
S/W	Arduino	<ul style="list-style-type: none"> <li>▷ clock, latch, data 신호의 제어.</li> <li>▷ LED Matrix의 상태 제어 및 회로의 전체적인 신호를 받아들여 작동 (줄의 제거, LED 신호, Score신호, 위치제어의 연산, 블록의 랜덤 출력, 작동중지)</li> </ul> 

[그림 2] 블록 모양

[표 1] 작품 규격

## 2. 설계제한 요소

[경제성]	① IC의 사용 개수를 최소한으로 하며, 사용 IC 내에서 활용도를 극대화한다.
	② IC의 사용 종류는 준비된 IC(실험 지원 IC)를 적극 활용하여 저예산으로 설계를 진행한다.
	③ LED Matrix 구현을 위해 Module을 활용하였다.
	④ 외관을 직접 제작한다.
[기술성]	① 다량의 LED를 병렬로 연결하여 원하는 x, y 축의 LED를 제어하며, 다른 LED들과 함께 빠른 속도로 점멸하여 원하는 모양을 시각적으로 나타나게 한다.
	② Shift Register를 이용하여 Digital I/O 핀 수를 원하는 개수로 확장한다.
	③ counter와 full adder, latch를 설계 및 연동하여 7-segment를 통하여 원하는 시간/점수를 시각적으로 표시 한다.
	④ 2축 조이스틱 신호를 받아들여 블록의 위치를 x, y축에 변경을 줘서 블록 위치 제어 한다.
	⑤ 최소 전압 5V를 준수하며, 안전성을 고려해 10V를 넘지 않는다.
	⑥ 1차 출품과 2차 출품으로 나뉘는 때, 2차 출품은 1차 출품의 발전 형태로 제작한다.

[표 2] 설계제한 요소

## 3. 설계방법 및 동작 원리

### [설계방법]

① Shift register	병렬 Shift register IC 2개를 이용하여 1번째 Shift register에 clock, latch, data 핀에 Arduino 코드에 따라 bit단위로 생성되고 보내진 신호를 받아들인 후에 출력되는 Serial OUT 출력 신호를 2번째 Shift register에 연동함으로써 필요한 Pin 확장 기술을 적용함.
② LED Matrix	Shift register에서 확장된 핀을 각 LED Matrix 1-24 pin(R/G color, used red color)에 연결하여 x축, y축에 해당하는 LED를 하나 ON한다. 이때, Arduino에서 사람이 인식하기 힘든 빠르기(50ms)의 속도로 점멸함으로써 원하는 모양의 블록 모양을 출력하게 한다.
③ 2-Axis Joystic	x축, y축 두 개의 신호 pin을 사용하여 위로 올릴 때 상태 값이 커지고, 아래로 내릴 때 상태 값이 작아지는 현상을 이용하여 움직인다는 신호를 받게 하여 블록의 위치를 제어한다.
④ 99-Counter	JK플립플롭 4개를 사용하여 1자리수(0-9)를 비동기식으로 만들고, 이를 연결하여 두 자리수(총 8개)로 만들고, NAND Gate를 사용하여 $1010(=10_{(10)})$ 일 때, 전체 초기화를 하여 다시 0으로 순환하도록 설계함.
⑤ 999-Counter	K플립플롭 4개를 사용하여 1자리수(0-9)를 비동기식으로 만들고, 이를 연결하여 세 자리수(총 12개)로 만들고, NAND Gate를 사용하여 $1010(=10_{(10)})$ 일 때, 전체 초기화를 하여 다시 0으로 순환하도록 설계함. 단, 99-Counter와는 다르게 4bit-latch와 4bit-full adder를 사용하여 Arduino에서 줄이 소거되는 신호를 받아들여 adder하여 latch에 저장하는 방식으로 설계함.

[표 3] 설계 방법

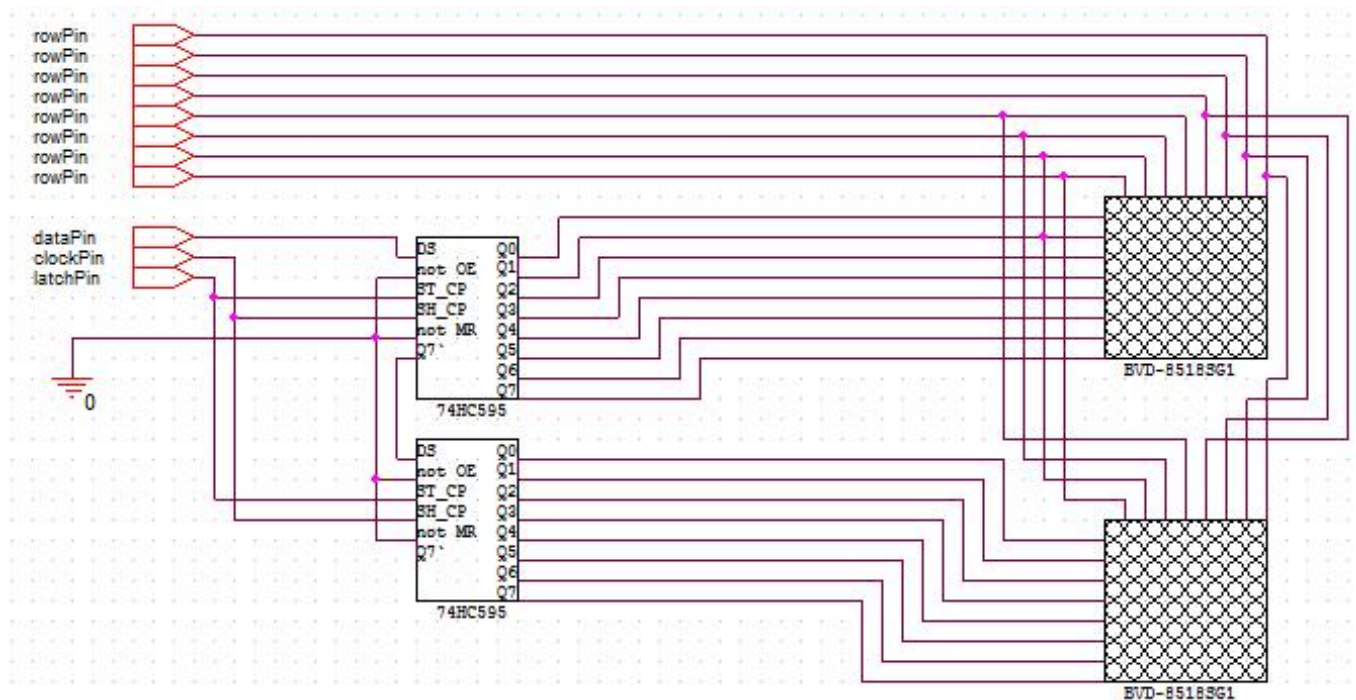
## [동작원리]

조 건		과 정				
<Arduino>						
Tetris 시작 & Joystic 제어	Arduino에서 연산	Shift register로 신호 분배	LED Matrix로 출력	999 Counter에서 가산	99 Counter에서 정지신호	정지
	1단계	2단계	3단계	4단계	5단계	6단계
<999-Counter>						
Tetris 시작	Reset	Arduino에서 score 전달	latch에 저장	full adder연산	다음 자릿수 연산	정지
	1단계	2단계	3단계	4단계	5단계	6단계
<99-Counter>						
Tetris 시작	Reset	1의 자리 Count	10의 자리 Count	90초 신호 출력	정지	
	1단계	2단계	3단계	4단계	5단계	

[표 4] 동작 원리

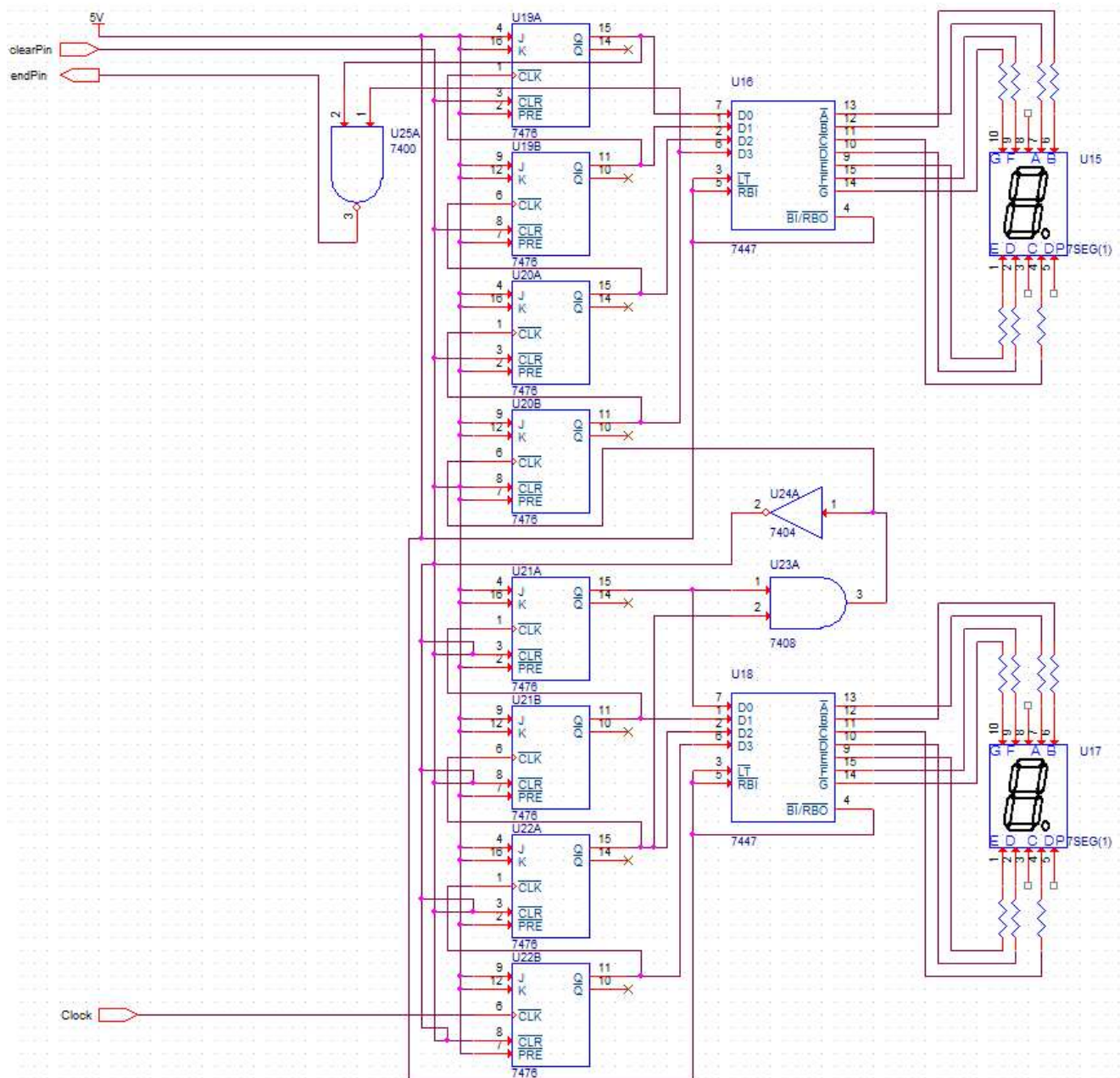
## 4. 설계도

### ·LED Matrix



[그림 3] LED Matrix

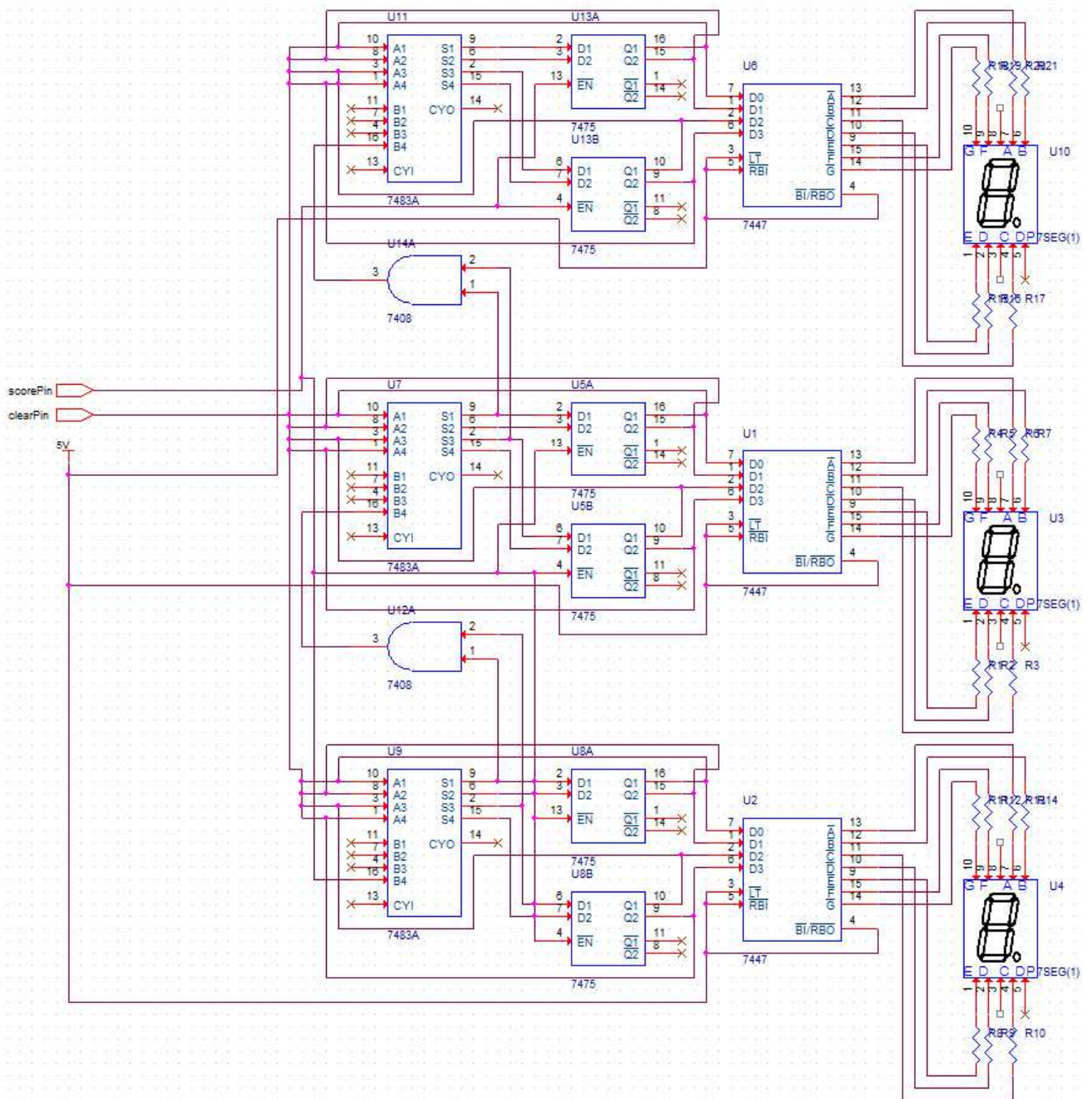
## ·99-Counter



[그림 4] 99-Counter



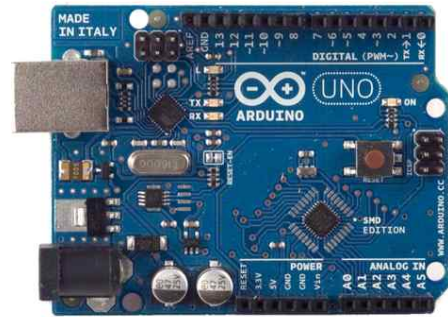
## ·999-Counter



[그림 5] 999-Counter

## 5. 부품

부 품	수 량 (EA)
Bread board	2 + 2(small) EA
8x8 LED Matrix (BVD-8518SG1)	2 EA
Analog 2-axis Joystic	1 EA
Shift register (74595)	2 EA
BCD Counter (7476)	2 * 2 EA
BCD to 7-segment (7447)	5 EA
Resistor(195Ω)	7 * 5 EA
4bit binary full adder (7483)	3 EA
4bit latch (74379)	3 EA
7-segment (FND507)	5 EA



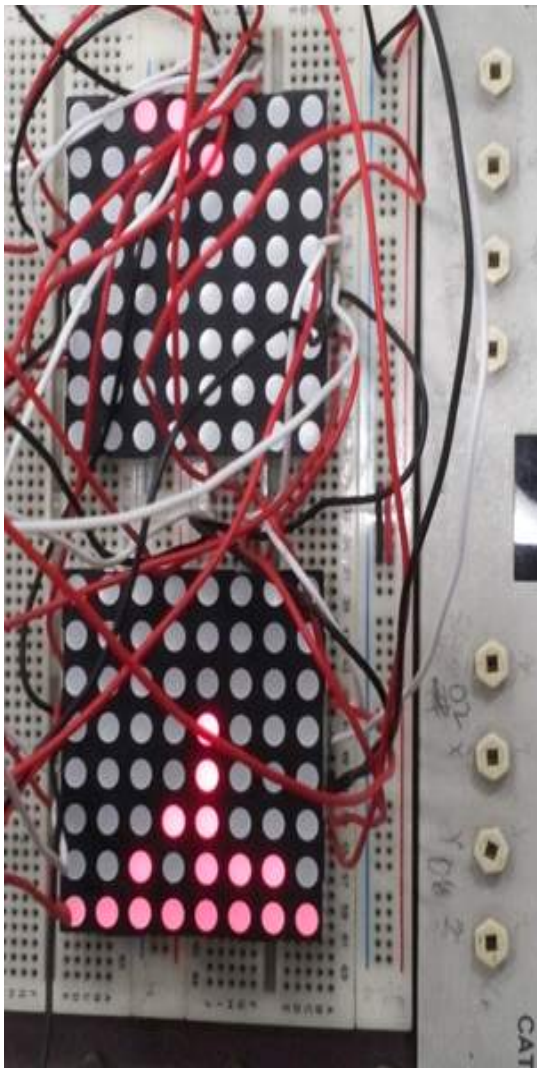
[그림 6] Arduino

[표 5] 부품

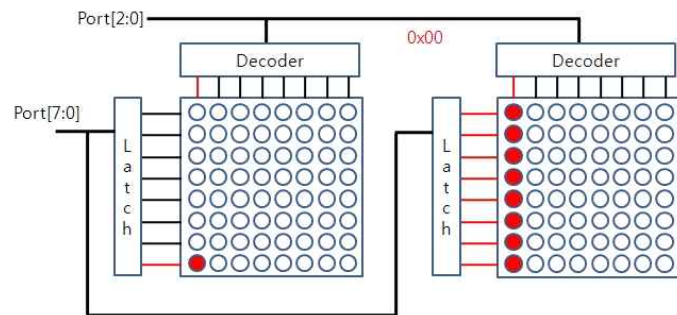
## 6. 구현 및 시험/평가

① Shift register를 이용한 LED Matrix 출력

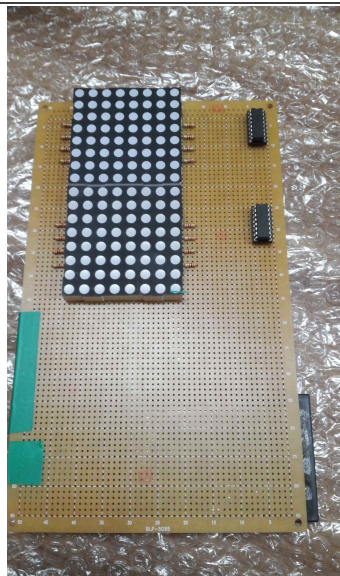
- 50ms의 점멸속도로 지정된 블록 모양을 순차적으로 출력하기 때문에 형태가 유지되어 보이는 잔상효과를 적용한 출력기술이다.



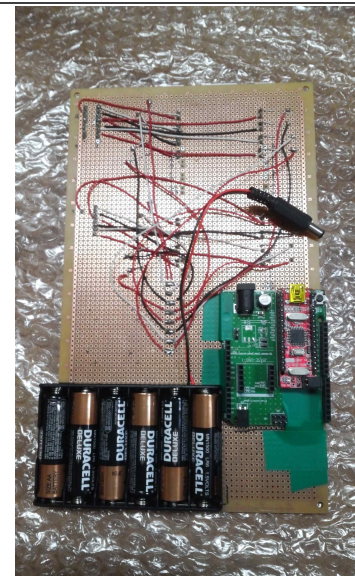
[그림 7] LED Matrix 출력



[그림 8] LED Matrix 동작원리



[그림 9] 납땜 - 앞

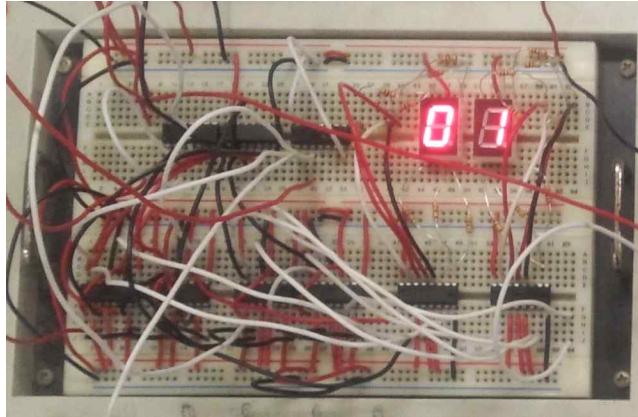


[그림 10] 납땜 - 뒤



## ② 99-Counter (Timer)

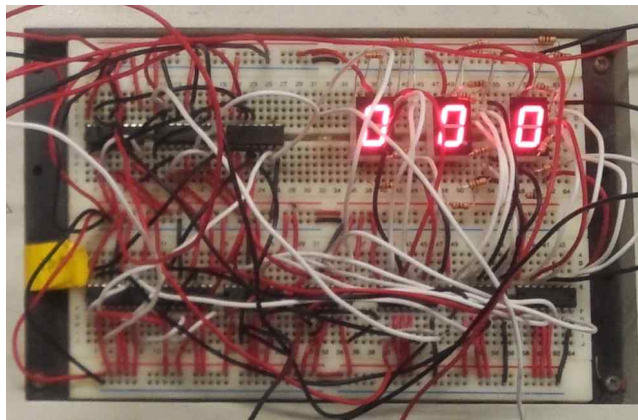
- 작동 시작 시 초기화를 하게 되는데, 처음 설계 시에는 각 자릿수마다 동시에 초기화를 했었는데 테스트 도중에 신호가 뒤죽박죽 섞이는 경우가 생겨서, 최상위 자릿수는 바로 초기화를 하고 다음 자릿수부터 지연시간을 주어서 신호가 차례로 전달되어 초기화되도록 설계함.



[그림 11] 99-Counter 구현

## ③ 999-Counter (Score)

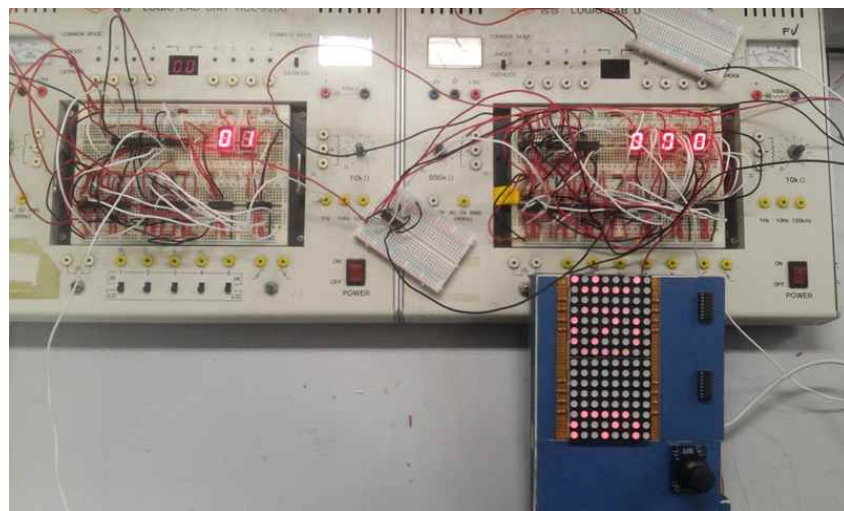
- 작동 시작 시 99-Counter와 같이 초기화를 하고 score 신호를 받아 latch에 저장을 한 뒤에 줄을 없애는 신호가 들어왔을 시에 가산하여 latch에 다시 저장하여 카운트한다.



[그림 12] 999-Counter 구현

## ④ 최종 연동

- 1차로 출품한 Tetris 본체에 회로를 추가 및 연동함.



[그림 13] 완성 작품



### ·진리표 J-K Flip Flop 진리표

\*J-K 플립플롭 (7476을 사용하여 UP Count 신호를 만들어서 Score와 Timer에 적용함), 비동기식으로 설계하였으며, 신호전달의 오류 때문에 지연시간을 주어서 신호를 재 정렬하였음.

입력		출력	
J	K	$Q_{t+1}$	$\overline{Q_{t+1}}$
0	0	$Q_{t+1}$	$\overline{Q_{t+1}}$
0	1	1	0
1	0	0	1
1	1	$\overline{Q_{t+1}}$	$Q_{t+1}$

[표 6] J-K플립플롭 진리표

### ·UP Counter 진리표

\*Score - 0~999 카운터 이므로, 100의 자리 1010과 10의 자리 1010과 1의 자리 1010 일 때 CLR 작동

\*Timer - 0~90 카운터 이므로, 10의 자리가 9가 되는 1001 일 때 CLR 작동

보유 하고 있는 JK-Flip Flop IC의 CLR pin이  $\overline{CLR}$ (0일 때 동작)이므로, NAND IC (7400)을 사용하여 CLR에 0을 주어야 작동을 한다.

논리식 :  $CLR = \overline{DC}$

입력			출력			
PR	CLR	CLK	D	C	B	A
0	0	↓	1	1	1	1
0	1	↓	1	1	1	1
1	0	↓	0	0	0	0
1	1	↓	0	0	0	1
1	1	↓	0	0	1	0
1	1	↓	0	0	1	1
1	1	↓	0	1	0	0
1	1	↓	0	1	0	1
1	1	↓	0	1	1	0
1	1	↓	0	1	1	1
1	1	↓	1	0	0	0
1	1	↓	1	0	0	1
1	1	↓	1	0	1	0

[표 7] Counter 진리표

## 7. 프로젝트 소감

김 철 언	초기의 목적은 자신이 배운 지식을 이용하여 자신의 위치를 파악하고 최초의 실무 작업과 함께 포트폴리오에 실적 하나를 써보고자 무작정 참여한 것이었는데, 직접 계획하고 제작에 참여하면서 무척 힘든 하루하루를 보냈다. 더군다나 기숙사에서 생활하느라 11시 전에는 돌아가야 했기 때문에 무척 초조했고 주말 시간을 투자해서 만들어야 했기 때문에 많이 지쳤었으며, 납땜을 통해 연동을 할 때 만들었던 부품이 죽어버리는 사태가 발생하여 더욱 괴로웠었다. 하지만 막바지에 이르러 다들 연동하거나 출품 때 작동이 안 되는 와중에 우리 작품은 완벽하게 목표로 했던 작동이 되어 무척 기뻐었다. 조장을 맡게 되어 못난 조장의 뉘달 속에 같이 함께 해준 조원들이 고맙다.
신 중 혁	팀원 간의 협동심이 얼마나 중요하고 우선시되어야 할 요소인지를 몸소 느끼게 되는 시간이 된 듯하다. 구성원간의 커뮤니케이션과 끊임없는 자기개발을 이뤄내며 팀원 한사람의 낙오자 없이 프로젝트 기간을 충실히 참여해야 하는 것의 의미를 다시금 되돌아보는 계기가 되었다. 프로젝트 기간이 기말시험 겹치는 바람에 팀원 간의 불화도 있었고 부품조달과 작품을 테스트하고 수정하는 단계에서도 애를 많이 먹었으며 작품이 완성된 마지막 순간에서까지 자잘한 예러 요소를 보이는 등 긴장을 늦추지 못하게 만든 프로젝트였지만 내 일생에 처음이었던 프로젝트를 의지가 되는 동료들과 함께 마지막까지 최선을 다해 원하던 결과를 만들어 낸 것에 대해 많은 보람과 즐거움이 되었던 시간이 되었다.
오 연 중	기존에 배운 논리회로 지식과 컴퓨터 기초설계시간에 사용하는 Arduino를 다루는 기술 등을 이용하여 Term Project에서 참여 하였기에, 자신의 실력을 확인함과 향상의 기회를 가질 수 있어서 좋은 경험이었다. 특히 주로 회로를 구성하는 역할을 맡았기에 조장의 요구에 맞춰 회로 작성함에 따라 회로도 보는 능력과 회로 구성 능력이 향상되어서 만족스런 기간이었다.
정 회 현	디지털 회로 설계시간에 이론 과 실습으로 배운 내용들을 직접 아이디어를 구상하여 작품을 제작하는 총체적인 과정을 토대로 더 심화된 회로 설계를 이해하였으며, 직접 회로작품을 설계하는 작업 자체가 참신하고 재미있었다. 비록 여러 번 벽에 부딪히기도 했지만 실험을 완전히 끝내고 되돌아보았을 때 이전보다 실력이 향상되었음을 실감할 수 있었다. 또한 Arduino를 이용하였기 때문에 소프트웨어적 요소뿐만 하드웨어와 연동하는 기술을 통해 실력을 향상시킬 수 있어서 좋은 기회였다.