

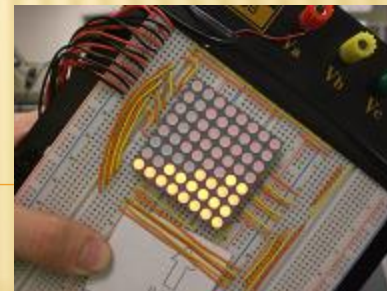
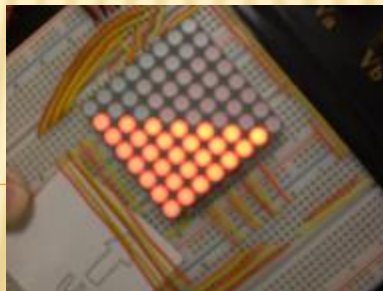
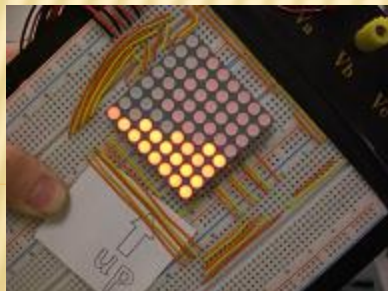
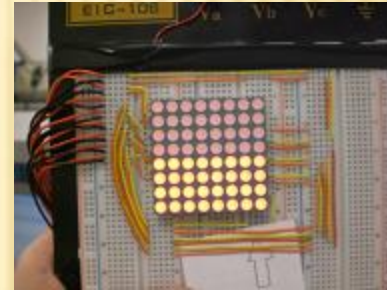
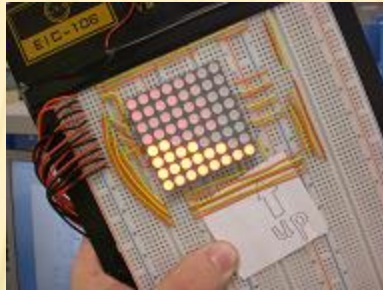
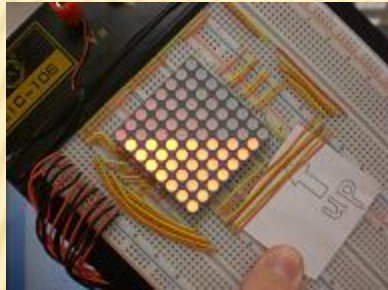
THE TRANQUIL TIDE MACHINE & LABYRINTH CMPEN 352W – FINAL PROJECT

Howie Doan Jonathan Georgino

April 28th, 2009

WE BUILT AN EMBEDDED SYSTEM...

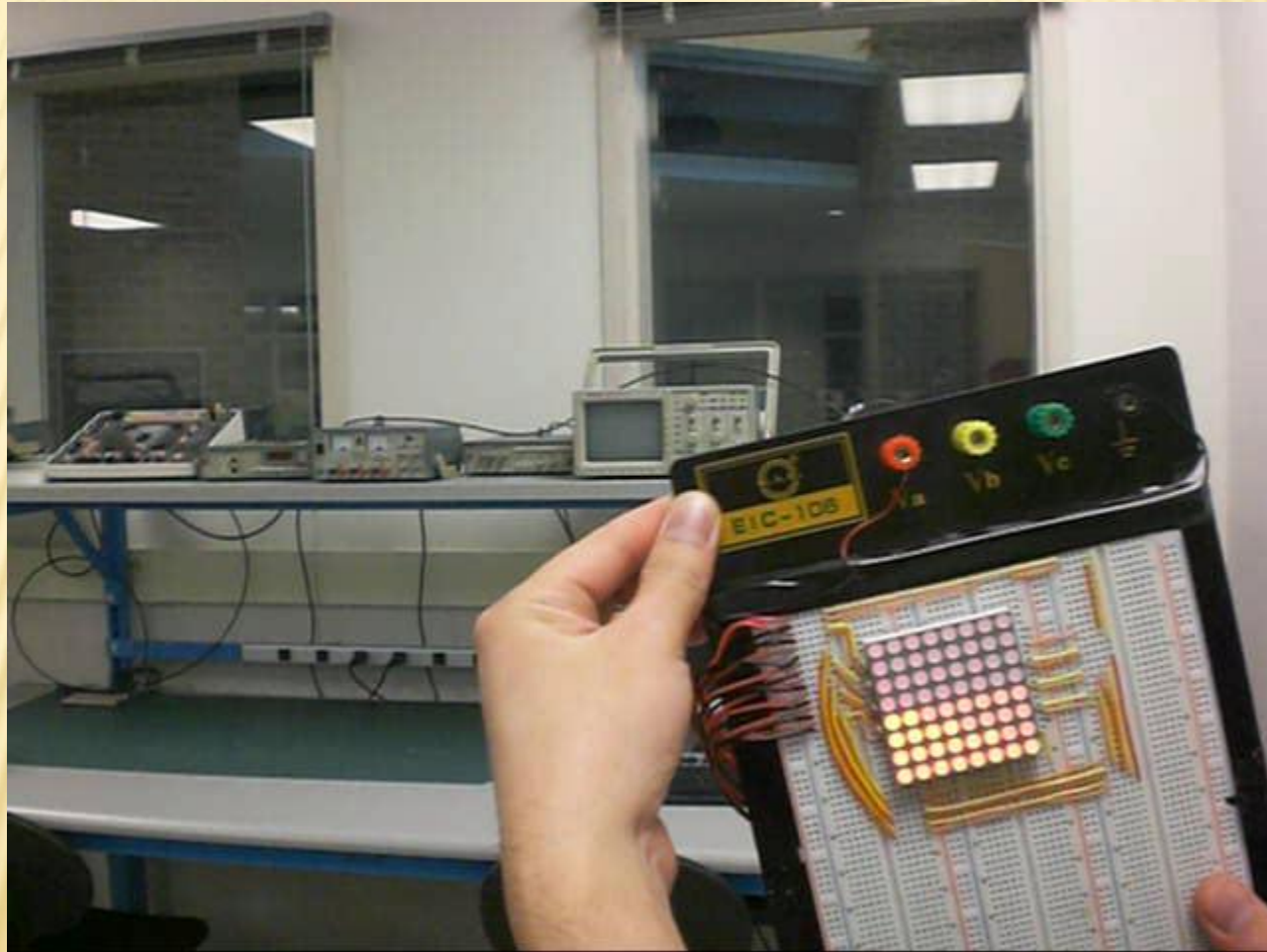
...which simulates light patterns and a labyrinth game for entertainment and to provide a relaxing experience for the user.



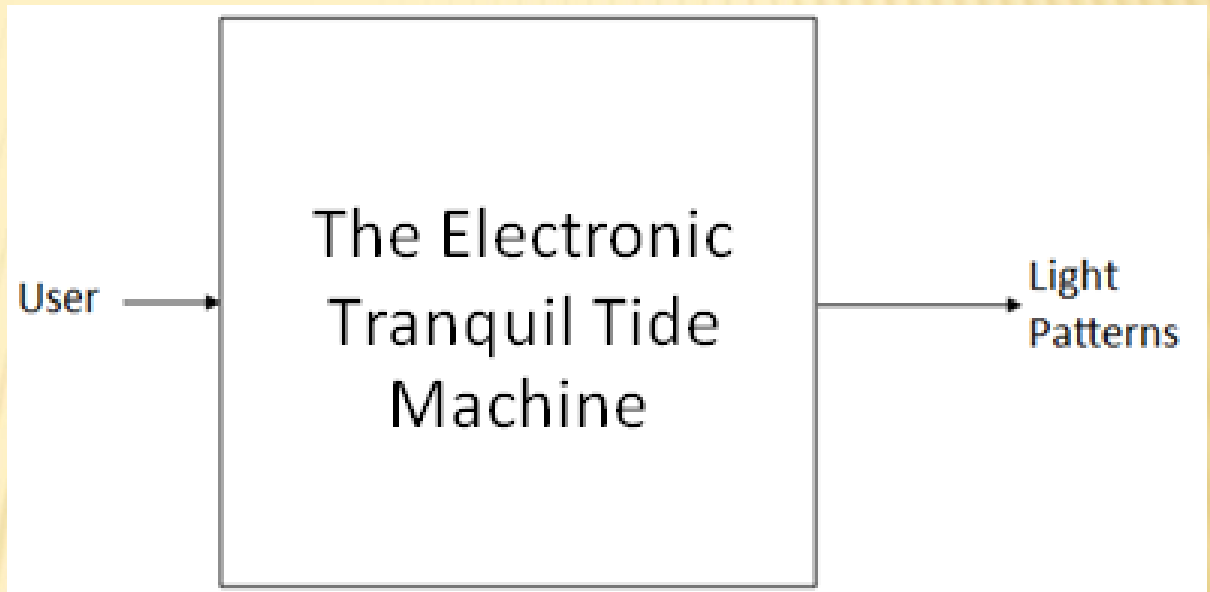
REQUIREMENTS

- ✘ Portable
- ✘ Easy to use
- ✘ Visible in typical home light conditions
- ✘ Relieves stress and entertains
- ✘ Movements based on actual physics calculations

DEMONSTRATION



LEVEL-0 DESIGN



INPUTS:

User
Environment

OUTPUT:

Light patterns

FUNTIONALITY:

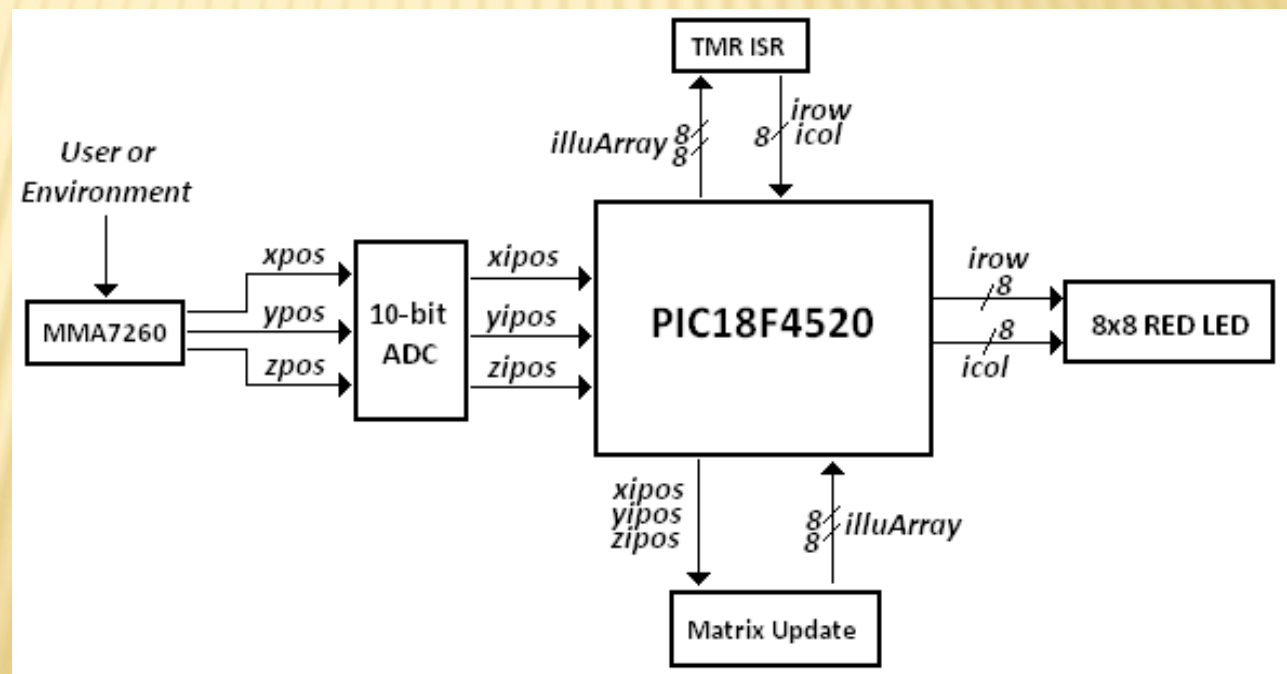
1. Tranquil Tide Machine
2. Labyrinth Game

LEVEL-1 DESIGN

- ✗ Input
 - + Accelerometer
 - + 10-bit ADC
 - + 3.3VDC

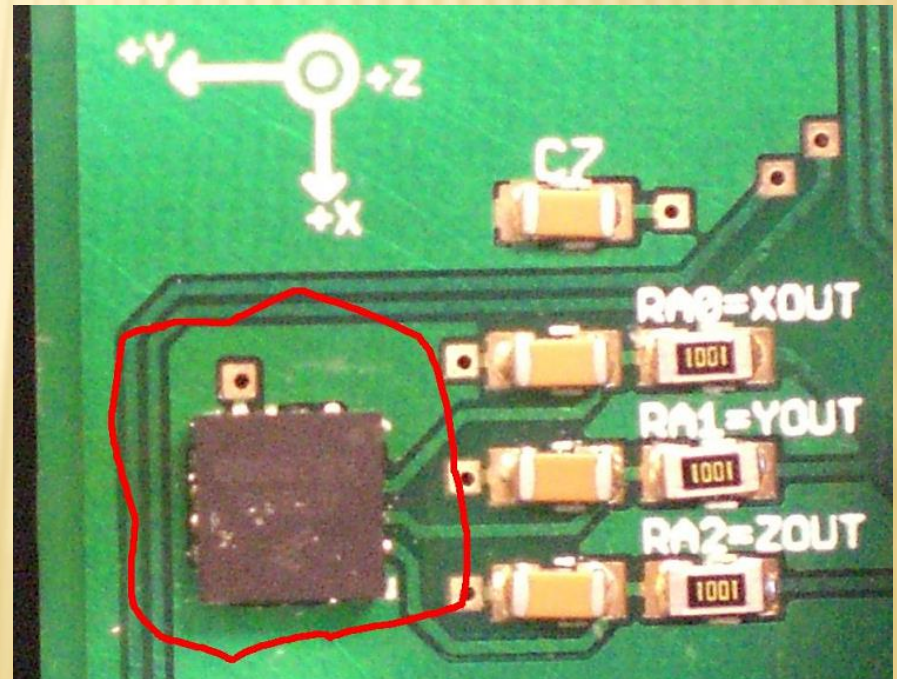
- ✗ Output
 - + 8x8 Red LEDs

- ✗ Processing modules
 - + TMRO ISR
 - + Matrix Update



LEVEL-2 DESIGN - ACCELEROMETER

- ✘ Part - MMA7260
- ✘ Input
 - + User/Environment
 - + Regulated 3.3V from Main Board
- ✘ Output
 - + x-axis 1.5g-scale analog output
 - + y-axis 1.5g-scale analog output
 - + z-axis 1.5g-scale analog output



LEVEL-2 10-BIT A/D CONVERTER

✘ Inputs:

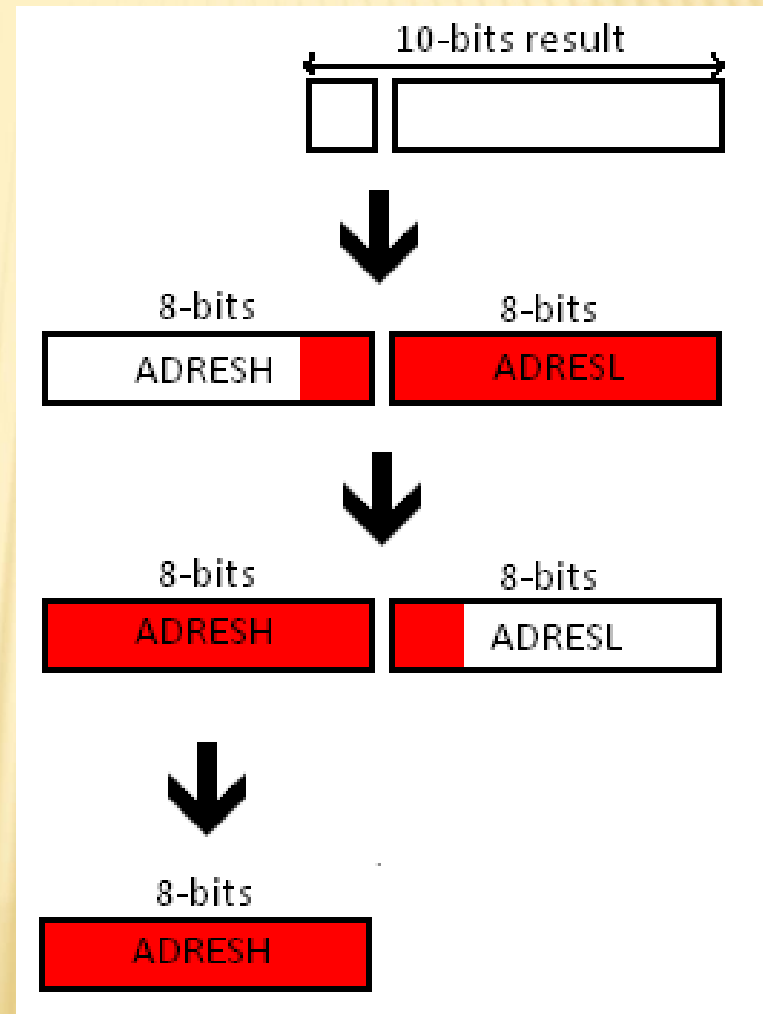
- + x-axis 1.5g-scale analog value
- + y-axis 1.5g-scale analog value
- + z-axis 1.5g-scale analog value

✘ Outputs:

- + x-axis digital value
- + y-axis digital value
- + z-axis digital value

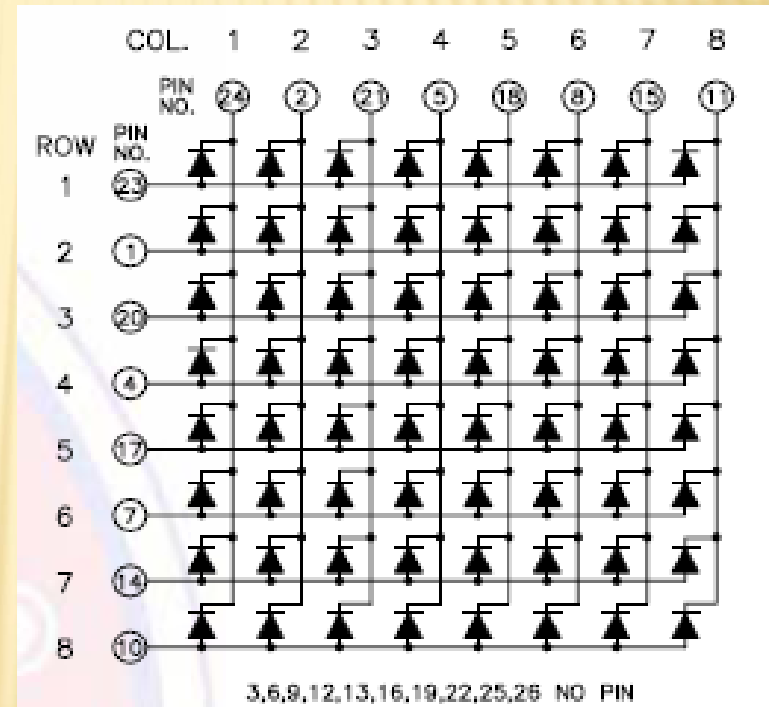
✘ Functionality:

- + One channel as a time



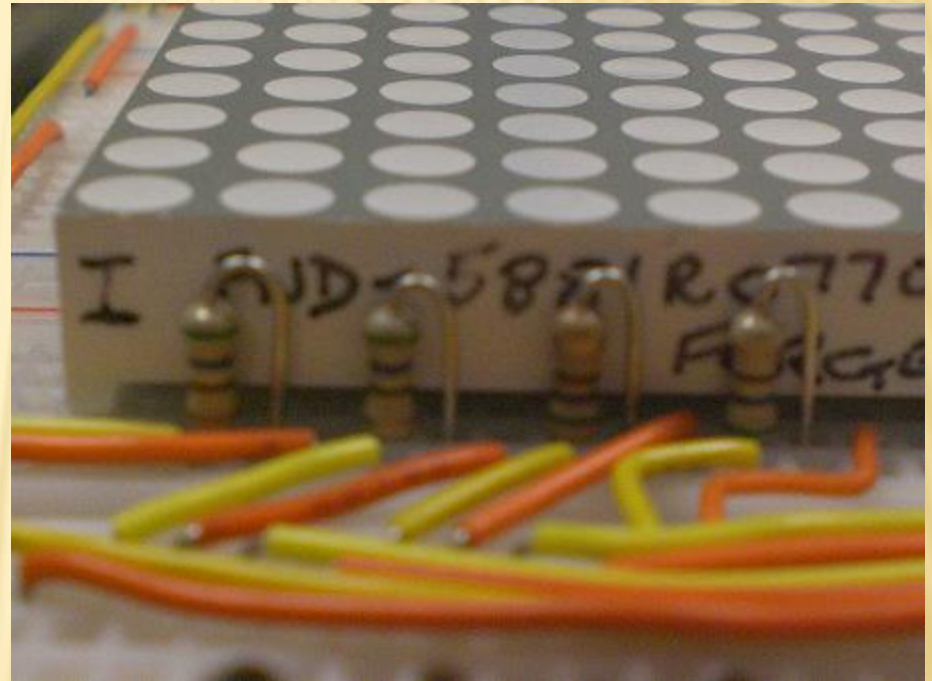
LEVEL-2 8X8 LED MATRIX FORGE EUROPA

- ✘ Part - FND-5881R0770CGC
- ✘ Inputs:
 - + 8-bit 1's hot encoded row select
 - + 8-bit column select signal
- ✘ Output:
 - + LED matrix



LEVEL-2 8X8 LED MATRIX FORGE EUROPA

- ✘ Current reduction resistor R value,
- ✘ $R = (V_{dd} - V_d) / I_d$
- ✘ $= (3.3 - 2)V / 25mA = 52\Omega$,
- ✘ where $V_{dd} = 3.3V$, LED forward voltage $V_d = 2V$, and LED forward current $I_d = 25mA$.
- ✘ Using 56Ω resistors (8)



THE PROCESS

- ✘ Collect Data from Accelerometer
 - + A/D conversion via on-board module
- ✘ Processing
 - + Matrix Updating
 - + TMRO ISR
- ✘ Output
 - + 8 x 8 LED Matrix

DETERMINING ACCELERATION

Acceleration	X-range	Y-range
+3	$\text{accelx} > 0xC0$	$\text{accely} > 0xC0$
+2	$0xB0 < \text{accelx} < 0xC0$	$0xB0 < \text{accely} < 0xC0$
+1	$0x84 < \text{accelx} < 0xB0$	$0x8F < \text{accely} < 0xB0$
0	$0x82 < \text{accelx} < 0x84$	$0x80 < \text{accely} < 0x8F$
-1	$0x70 < \text{accelx} < 0x82$	$0x75 < \text{accely} < 0x80$
-2	$0x65 < \text{accelx} < 0x70$	$0x60 < \text{accely} < 0x75$
-3	$\text{accelx} < 0x65$	$\text{accely} < 0x60$

DETERMINING VELOCITY AND POSITION

- ✘ Velocity = velocity + acceleration
- ✘ Terminal Velocity = 0x10 (all directions)
- ✘ If(velocity < 0x02)
Position = position + 1
Else if (velocity > 0x02 && velocity < 0x04)
Position = position + 2
...If(velocity < 0x02)

MYSTERIOUS HOLE & COLLISION DETECTION

- × Implemented by using a 10x10 array

```
char universe[10][10] = {'W', 'W', 'W', 'W', 'W', 'W', 'W', 'W', 'W', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'M', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'W',  
                          'W', 'W', 'W', 'W', 'W', 'W', 'W', 'W', 'W', 'W'};
```


FURTHER IMPLEMENTATION

- ✘ Walls
- ✘ Z-Axis
- ✘ Duty cycle
 - + Pulsing
 - + Dimming
- ✘ More balls