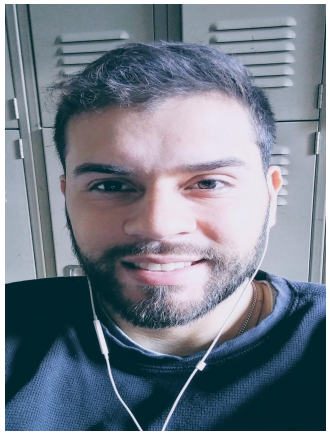
Acoustic Battleship
Team 5 MDR
12/11/2019

Evaluated by:
Professor Maciej Ciesielski
Professor Yeonsik Noh
Professor Baird Soules

Team Members

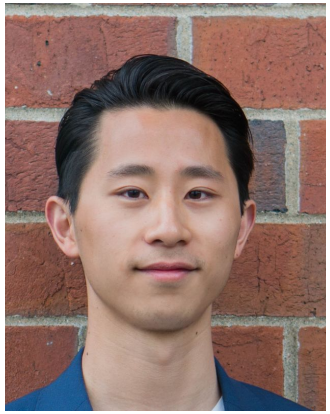
Liam Weston
(Group Manager)

Adrian Sanmiguel



Justin Forgue

Xinyu Cao



Problem Statement

Board games have failed to adapt to the technological advances of today's market. Traditional board games have fallen out of favor. Implementing embedded systems could help to provide a jolt to the industry.

How do we plan to do this?

Problem Statement

- Provide an aesthetically pleasing, functional, scalable, and robust interface
- Applying these characteristics to Battleship

Problem Statement

- Our solution will put an interactive spin on a classic game
- Accuracy based game using a ping pong ball to provide low-latency, responsive feedback
- Will follow an adapted set of guidelines to Battleship
- Using localization from a network of microphones to detect if a target is hit

Game Rules

- Two team game (1+ player per team), alternate turns
- 1m x 2m playing surface
- projectile is a ping pong ball
- Each team is attempting to hit multi-coordinate, line of sight platform, where a coordinate may contain a target
- A winner is declared when all targets on either team have been struck by the opposing teams projectile

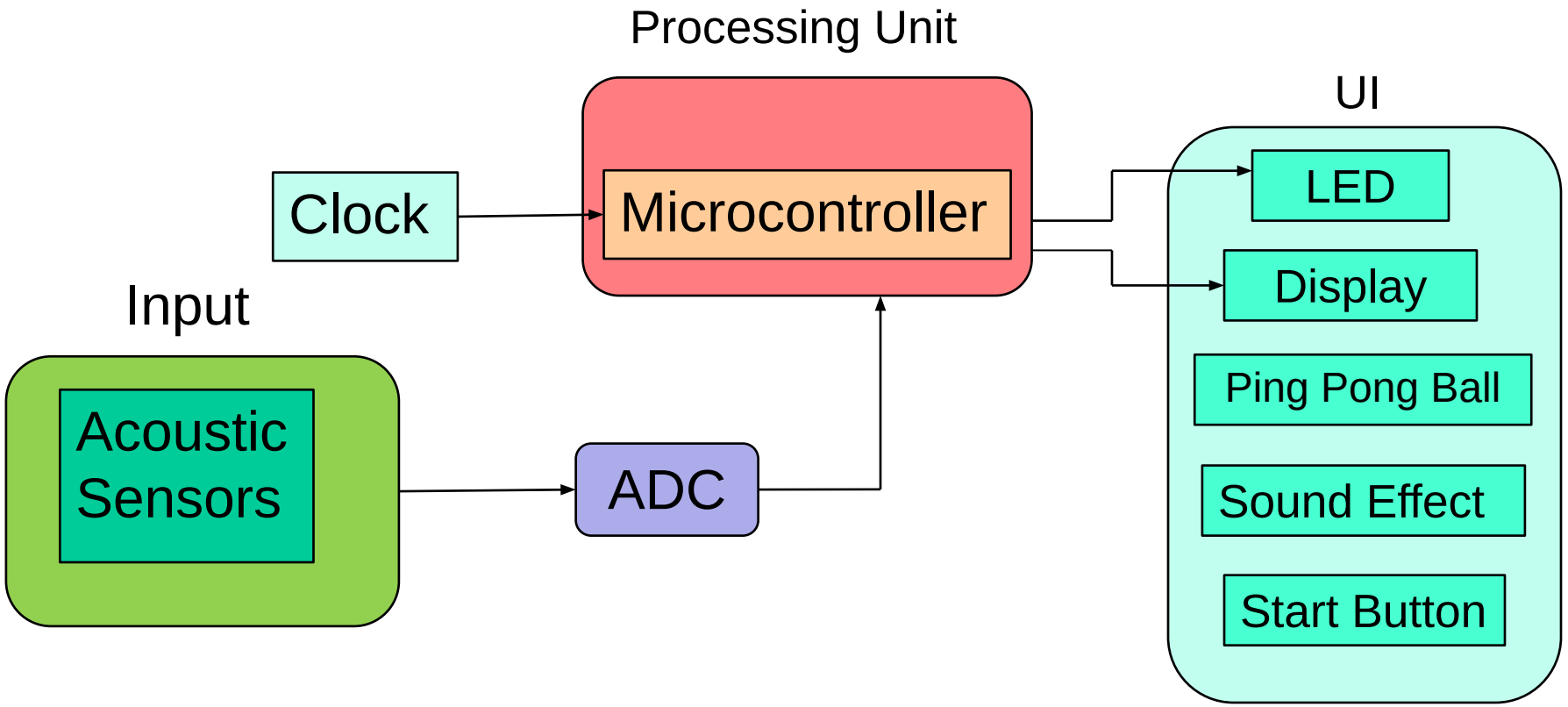
System Requirements & Specifications

Table of Requirements and Specifications

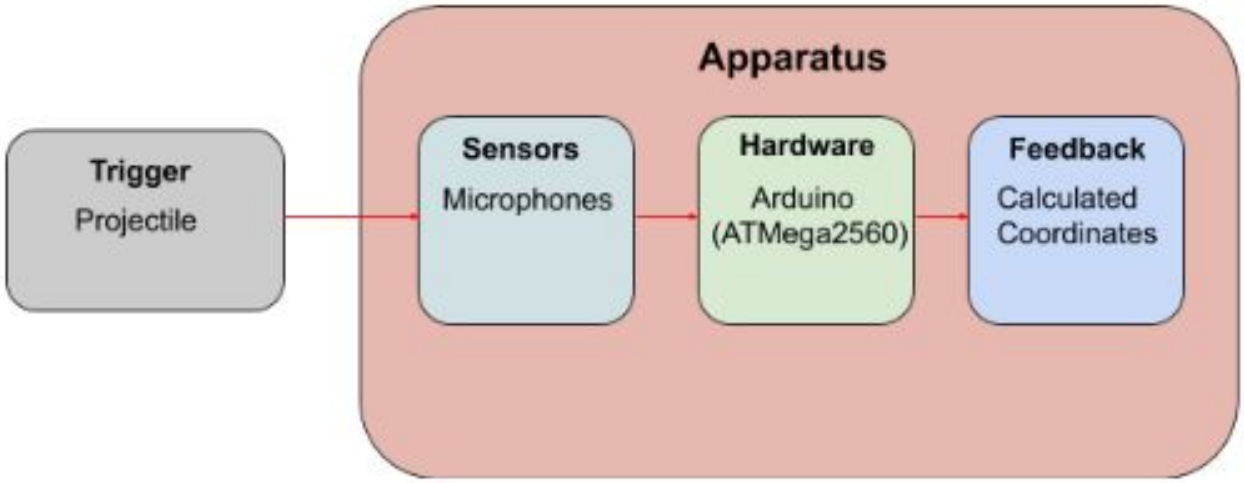
Requirement	Specifications	Value
Accuracy	Distance Error	<= 5 cm
Responsiveness	Response Time	<= 500 ms

Components:
Microphone, LED, ADC, Microcontroller, Ping-Pong Ball, transparent playing surface/Display

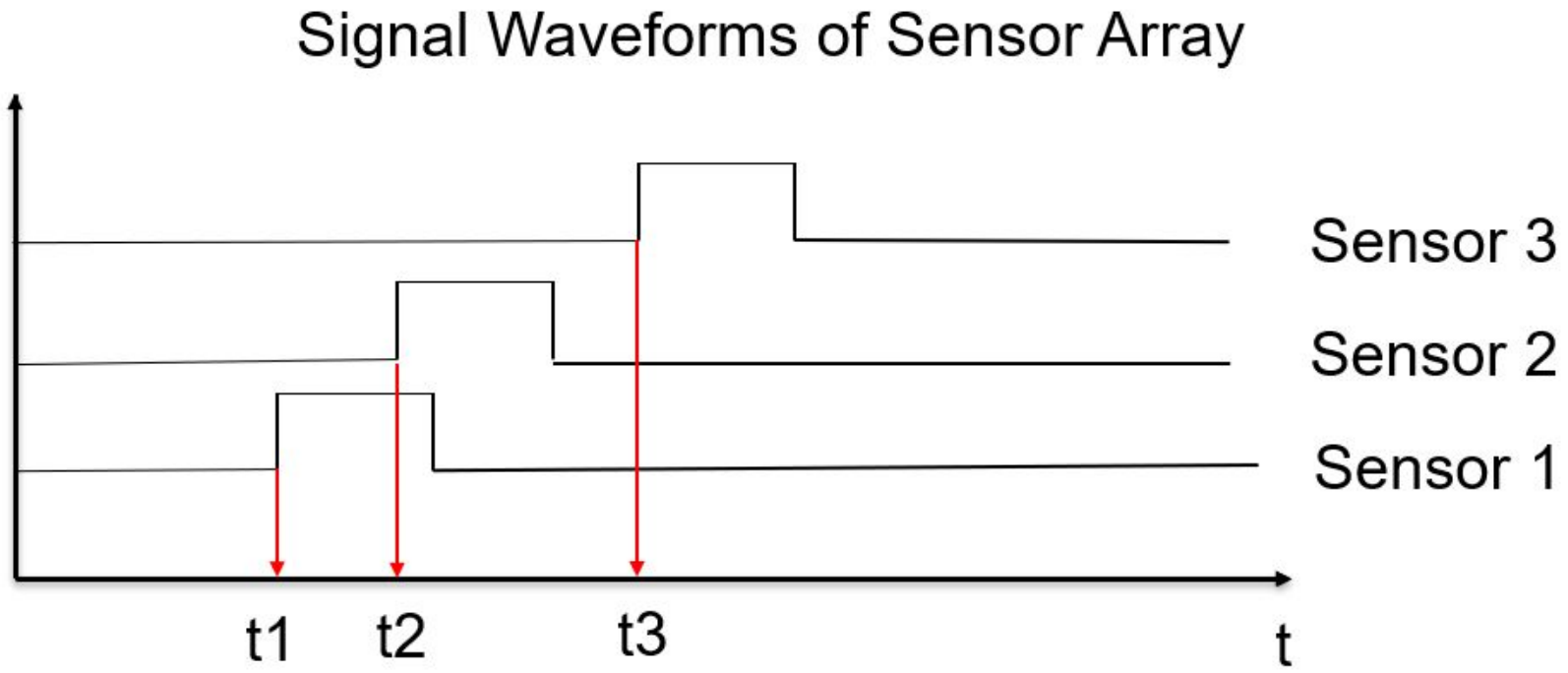
Block Diagram (PDR)



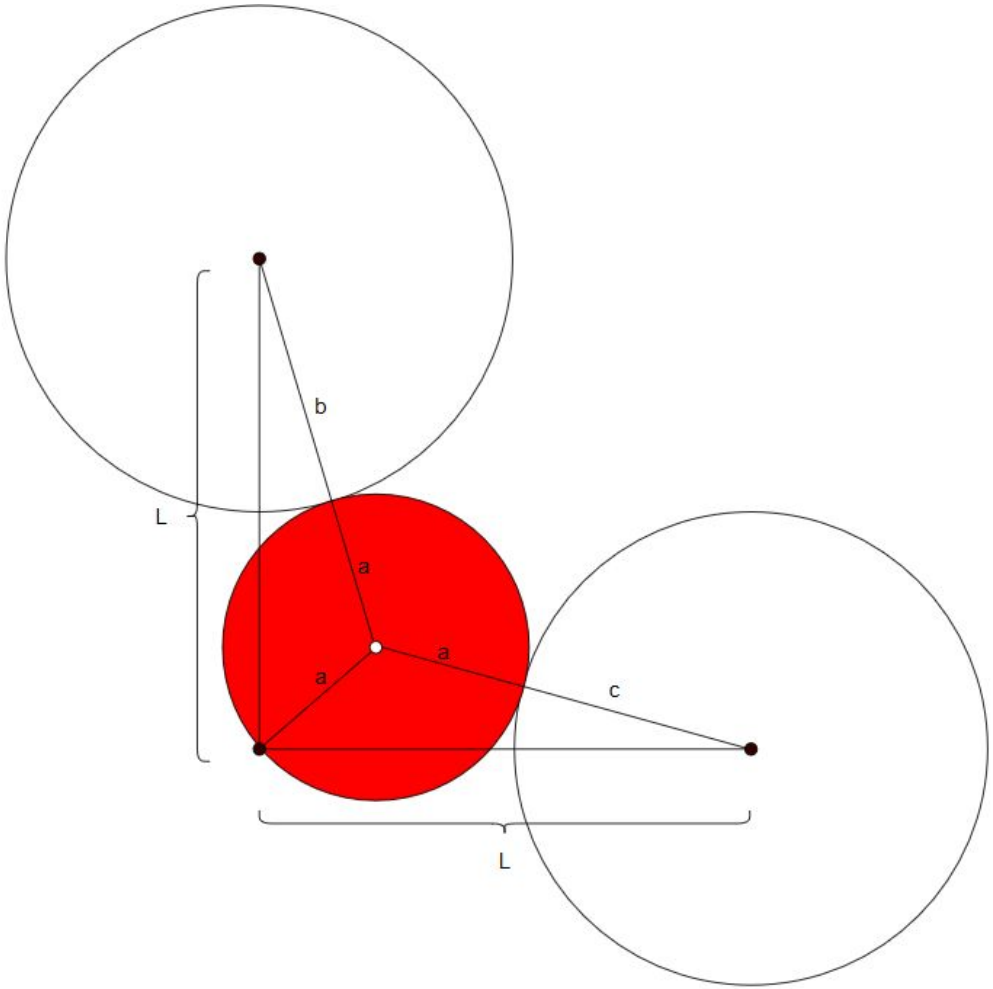
Block Diagram (MDR)



Time of Arrival



Algorithm



Known Values

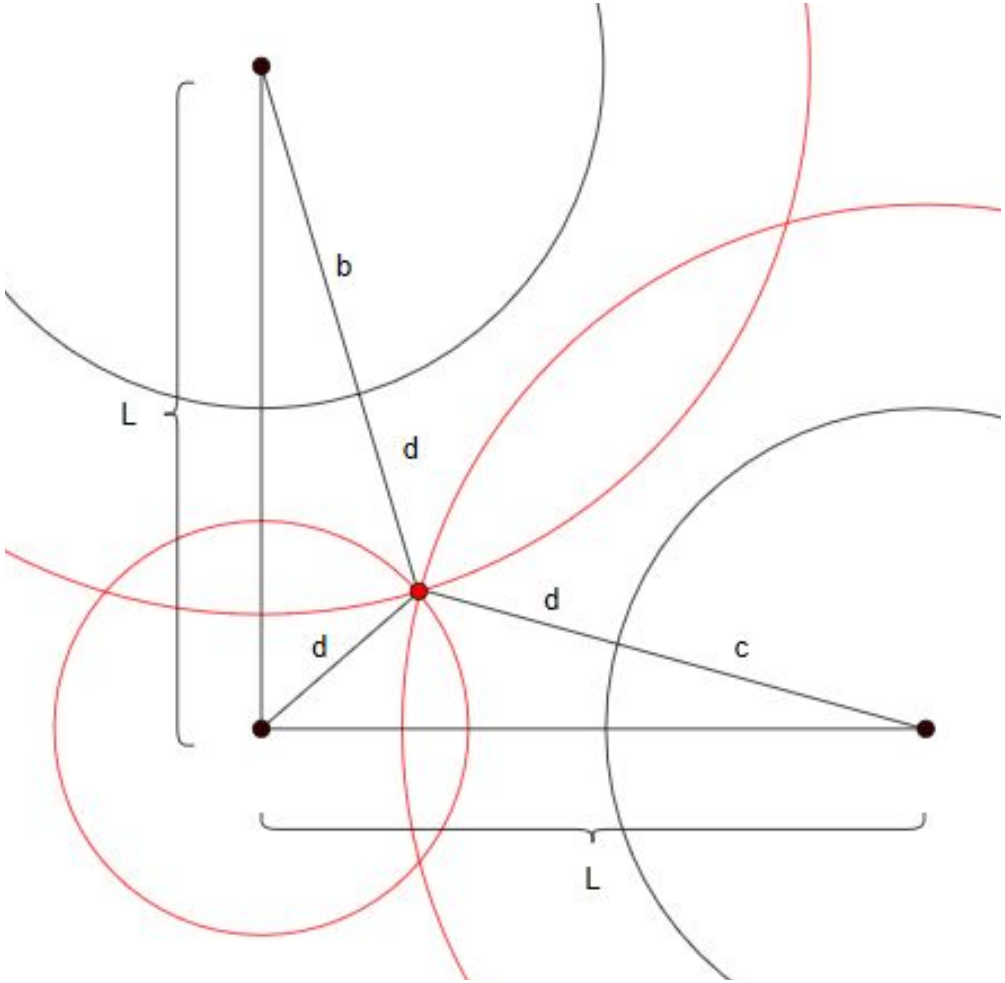
- $b = \Delta t_b \times 343\ {}^{m}/_{sec}$
- $c = \Delta t_c \times 343\ {}^{m}/_{sec}$

We must calculate the distance of a based on b and c

Steps for Analytical Solution

1. Send distance **a** to zero
2. Create new circle with radius **d**
3. Increase **d**, simultaneously increase radii **b** and **c** by **d**
4. When the three circles intersect at a unique point, we have determined the source of the sound.

Algorithm Continued...

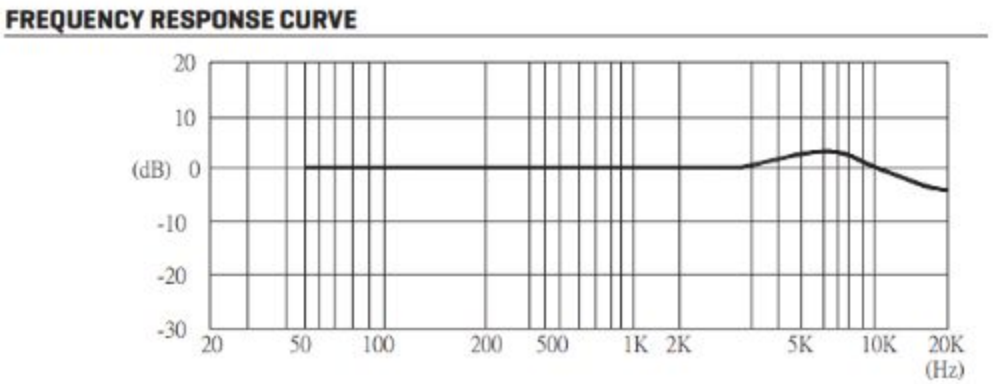


Microphone Sensors

- Implement 8 electret omnidirectional condenser microphones (CMA-4544PF) to optimize source localization in 2-Dimensional space
- Operating frequency: 20Hz – 20kHz

  Frequency of human conversation: 85Hz - 255 Hz

  Frequency of Ping Pong hitting a surface: 5.9kHz - 7.3kHz



CMA-4544PF

Sensor Hardware (MDR)

Electret Condenser Microphone

- Automatic Gain Control
- Low - noise microphone bias
- Variable gain: 40,50,60 dB
- DC offset: 1.25 volts

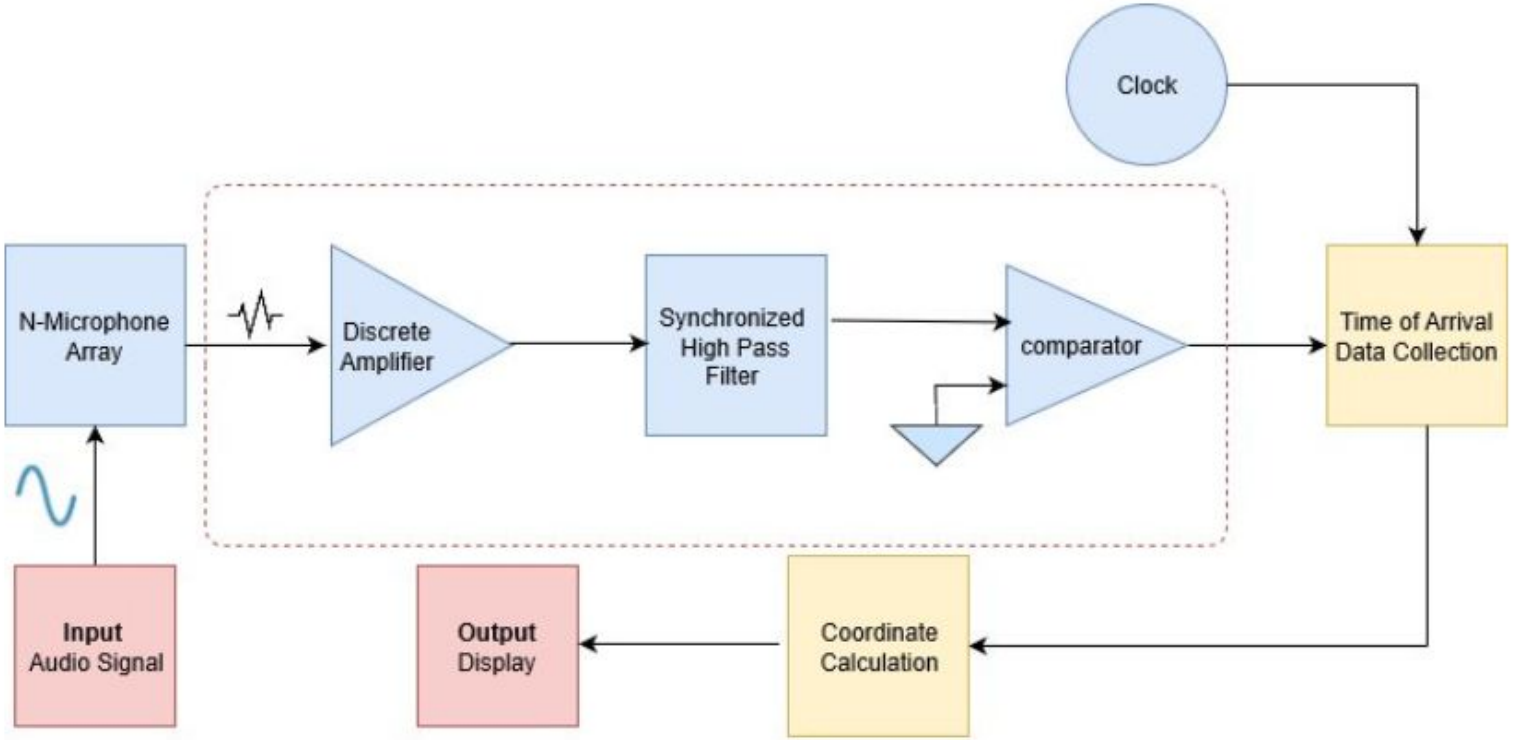
Adafruit AGC Electret Microphone - MAX9814

Schmitt Trigger Inverter

- Comparator with Hysteresis
- Two threshold voltages
- Used to provide a digital high-to-low output from each microphone sensor

74HC14N IC

Analog Digital Converter (ADC)



Microcontroller

Arduino Atmega2560

- 16 MHz clocked prescaled at 250 kHZ
- ~ 2mm resolution
- Four 16-bit synchronous timers

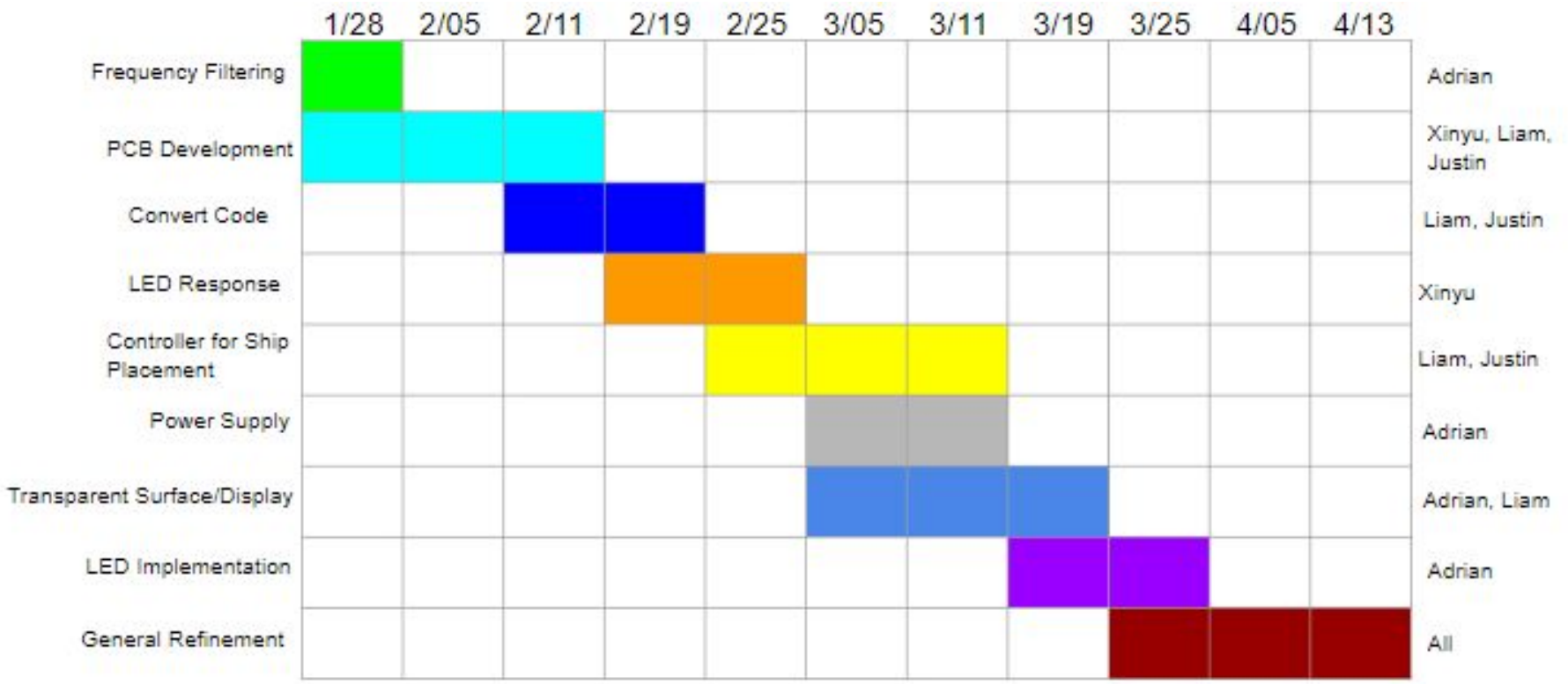
74HC08

- Four AND gates/sensor used as delay

Microcontroller Function

- Takes input from the ADCs and clock
- Once the input of an ADC goes from high-to-low the system time is stored
- The 8 time stamps are compared to calculate a location on the board
- The location is matched to a LED
- The relevant LED is switched through the output of a PWM signal

Gantt Chart


1/28
2/05
2/11
2/19
2/25
3/05
3/11
3/19
3/25
4/05
4/13
Frequency Filtering
Adrian
PCB Development
Xinyu, Liam, Justin
Convert Code
Liam, Justin
LED Response
Xinyu
Controller for Ship Placement
Liam, Justin
Power Supply
Adrian
Transparent Surface/Display
Adrian, Liam
LED Implementation
Adrian
General Refinement
All

MDR Prototype (Original)

- System on a single board for one player
- Using Arduino as microcontroller
- Calculate coordinates and light up LED accordingly
- Error distance less than 8 cm.
- Response time less than 1 s

MDR Prototype (Actual)

- System on a single playing surface for one player
- Use an Arduino Mega2560
- Response time less than 500ms
- Error distance less than 8 cm.

CDR Game Rules

- Players choose the positions for their battleship through certain type of controller; positions are displayed on the LEDs board visible to each player themselves
- Players attack their opponent in turn by throwing ping pong ball at their opponent's surface
- Players score when they hit the battleship, as indicated by the LEDs on the surface
- The one who hits all the battleships first wins

CDR Interfaces & Specifications

- Two transparent square surfaces for players
  - LEDs under the surface show and register hit or miss
  - 10 * 10 block on each surface
  - 1 meter * 1 meter on each surface
  - Four sensors placed one at each corner
  - A button for each player to push to indicate turns
- Sub LED Displays
  - controller to select battleship position individually
  - small display of LEDs displaying battleship positions
- Specifications
  - Response time less than 500ms
  - Distance error less than 5 cm
  - User friendly game experience

Questions?