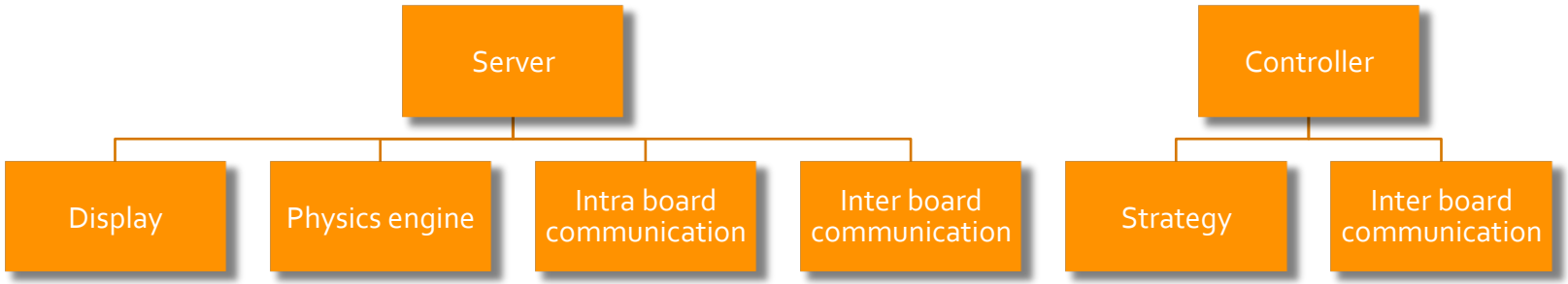
Real-time Soccer Controller

EE4214 – Real time embedded systems

Introduction

- Real time soccer controller for 5 a side team



Server – Display

- Renders GUI on the TFT
- Communicates with physics engine via mailbox
- Data packets received from the physics engine trigger UI handlers

- Main display states are pre and in game frame
  - Pre-game frame
    - Triggers UI changes after receiving initial player position packets from both team
  - During game frame
    - Draw field to render player and ball movement
    - Update score board and timer

Server – Display

- Implementation
  - Use of 3 different memory stacks as video memory
    - One for Pre Game Frame
    - Two for In-Game Frame
  - Reducing computation to draw ball player by using pre-loaded pixel array
  - Drawing circle efficiently using Bresenham Algorithm (Raster Cirlce)
  - Use of double buffering to reduce flicker and increate frame rate
- Road Block
  - Bottle neck due to Set_Pixel and Get_Pixel system calls
    - Reduce computation speed, hence refresh rate affected
    - Solution : Reduce code size and make it fit inside the BRAM, or generate new hardware such that the entire BRAM space of 64K is allocated to Microbalze carrying Display Code.

Server – Physics Engine

- Wall collisions – horizontal and vertical
- Triangle edge collision
  - Equations of four edge (shifted) lines used

    x = -y and
  - Avoided square roots and other components that require intensive processing

Server – Physics Engine

- Player-Player collision
  - Find the angle between players using tan inverse
  - Reverse the overlap of players based on granularity
  - Exchange components of velocity along direction of collision
- Ball-Player collision
  - Find the angle between players using tan inverse
  - Reverse the overlap of ball-player based on granularity
  - Component of velocity along collision line made negative

Inter board communication

- Data encoded as 32 bit packets
- UART Interrupts to receive data
- Data decoded every clock cycle
- Decoded data stored in circular queue buffer
  - UART buffer size limited
  - Interrupts are kept smaller
  - Queue size default 25 packets

Other components and functions

- Mailbox
- Push buttons
- Timer
- State machine to select execution of commands
- Spot refereeing
- Random generation of players

Controller - Strategy

- 3 types of players – goalie, attacker, defender
- Goalie – guards the goal by tracking y coordinate of ball
- Attacker – 2 attackers
  - One attacks ball – attacker closest to ball
  - Other helps first attacker
- Defender – 2 defenders
  - Both man mark players closest to them
- Dynamic strategy change – offensive and defensive

Problems Encountered

- Communication problems
- Delayed testing process (lack of incremental coding)
- Inaccuracies associated with mathematical calculations
- Improper use of data types (fixed point, double and other notations)
- Conversion of data between multiple systems (px/s -> bit notation, radians -> degrees)

Scope for improvement

- Display code in BRAM
- Handling packet loss
- Handling overlaps better
- Reduce use of global variables to reduce BSS

Demo