

Tic-Tac-Tile

Lucas Jung [@gruvw](mailto:luc.jung@gruvw.ch)

01 / 2024

Microprogrammed Embedded Systems

NDS features: checklist (1/4)

■ **ARM Processors**

- ARM9 to control the two screens, the buttons (Left, Right, Up, Down, A, Start, Select), send IPC messages to ARM7 to control sound and Wi-Fi.
- ARM7 for audio, touchscreen and Wi-Fi.

■ **Timers / Interrupts**

- Timer 0: variable speed, game time progress bar.
- Timer 1: `TIMER_FREQ_64(60)`, game state FSM update.
- Timer 2: `TIMER_FREQ_64(10)`, button debounce, Wi-Fi spam prevention.

NDS features: checklist (2/4)

■ Graphics

- Main display: Mode 5 with backgrounds 2 and 3 (ext. rotoscale), used to display the game pieces, and the board background + begin menu.
- Sub display: Mode 5 with backgrounds 0 (tiled mode) and 2 (ext. rotoscale), used to display the settings selection + progress bar, and the game over screen.

■ Keypad

- Controlling the top game board actions (interrupts):
 - Left, Right, Up, Down = Move selected cell
 - A = Place piece on selected cell
 - Start = Start, Terminate, Restart game
 - Select = Reinitiate NDS pairing (Wi-Fi)

NDS features: checklist (3/4)

■ Touchscreen

- Select the game settings in the game menu, 6 different touch areas used, select game difficulty (speed) and game mode.
- Reading touchscreen by polling.

■ Sound

- There is a background music (MOD format) in the menu.
- Another background music (MOD format) is playing during a game. It will speed up depending on the selected game difficulty.
- Playing 3 sound effects (WAV format): when selecting a setting in the menu, when placing a piece on the board, and at the end of the game.

NDS features: checklist (4/4)

■ Sprites

- 7 sprites in the project: 4 sprites are used to display the outcome of the game on the Game Over screen: (1) crown for the winner, each side (2) has their sprite, (1) clock to indicate losing because of timer; 3 sprites indicate the status of the NDS connection/pairing (Wi-Fi).

■ Wi-Fi

- Wi-Fi connection with AP is used to communicate between two NDS and play two player games.
- Designed and implemented P2P-BOP protocol to pair two NDS.
- Designed and implemented Message Queue and ACK system to avoid packet loss.
- Added simultaneous message (START) action recovery mechanism.

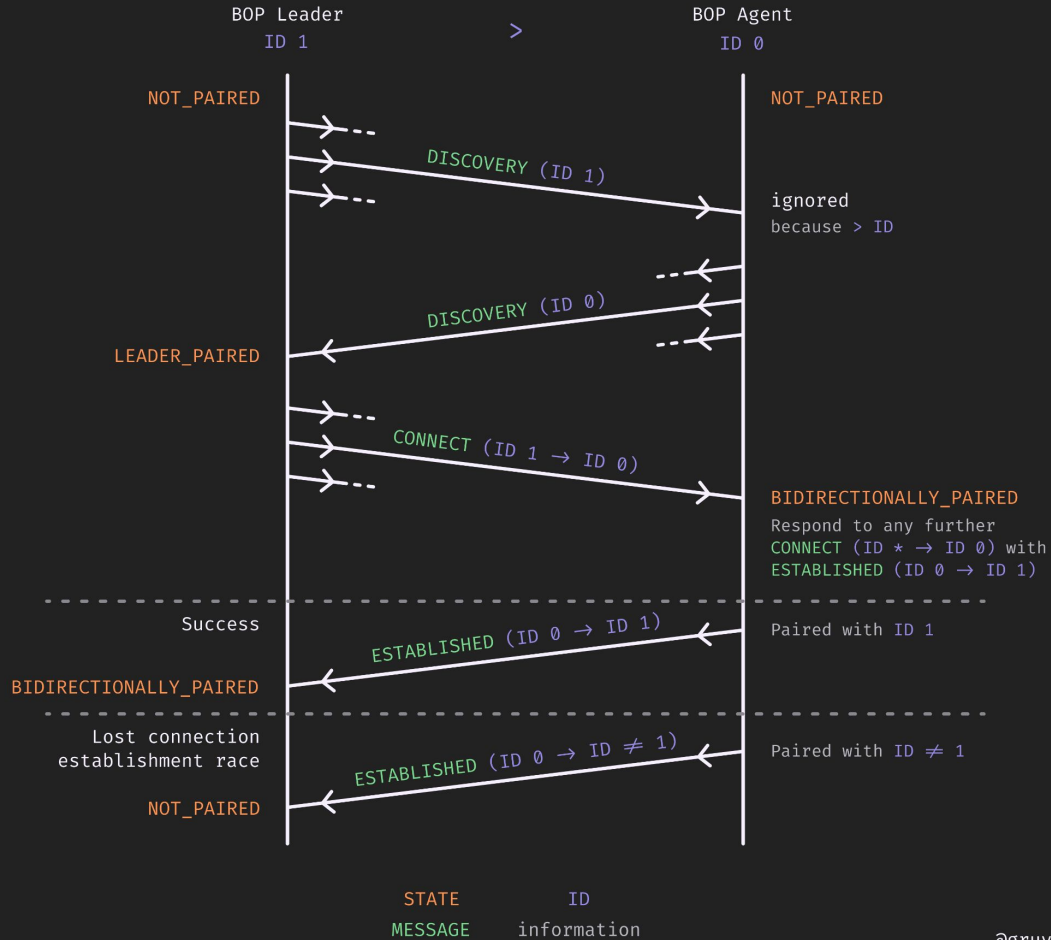
See project **README** for more details!

NDS improvements

- Implemented a bot/AI for single player mode that plays perfectly: using a minimax algorithm to find the best move + hard coded every first move response as it was too slow (~2 seconds) to compute.
- Added button debounce procedure (using a timer) to avoid double trigger of NDS physical buttons.
- Modified WiFi_minilib: implemented Wi-Fi AP connection timeout (do not stuck the game if can't connect to AP, can still play other game modes even after connection failure).
- If both players press START at the same time (in Wi-Fi mode) it will launch a conflict resolution process and still work as expected.
- Wi-Fi stack implementation stands very strongly against packet loss thanks to the ACK and Message Queue system.

See **source code** (close to 2000 lines) for more details!

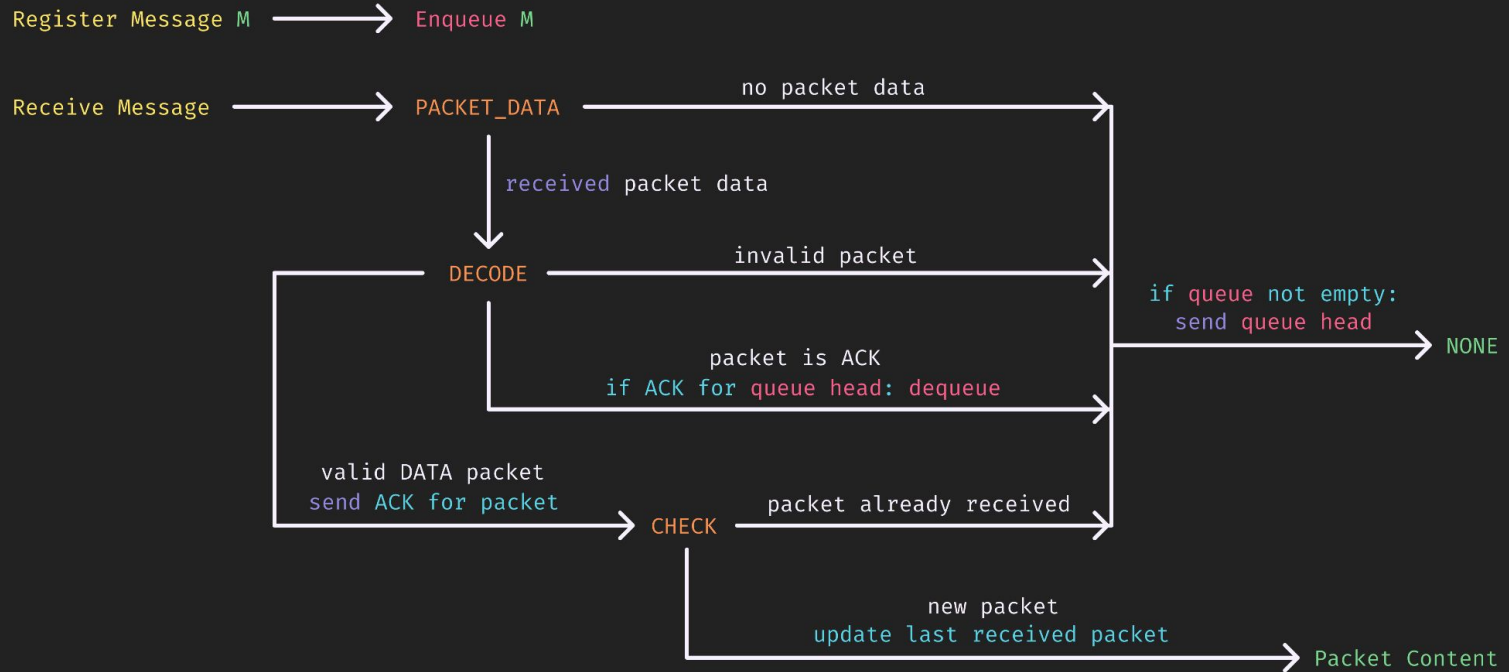
Peer-to-Peer Broadcast Ordered Pairing (P2P-BOP) protocol



Message Queue and ACK system FSM

Assumes P2P-BOP state `BIDIRECTIONALLY_PAIRIED`

Ambient Fixed Size Message Queue



STATE action condition interface
network message queue information

NDS project screenshot

