Rehabilitation System for Stroke Recovery: Real-Time on RA8D1

Visual Finger Flex Simulator (TRON Competition)

Renesas RA8D1 TRON Competition

ENTRY NO.: 22536

MEMBERS:

JOSTIN JAISON

KEERTHANA N

Redefining Recovery. Rewiring Motion.

- **Problem**: Port a complex **Haptic Glove System** into a purely software simulation on the Renesas RA8D1 MCU.
- **Challenges**:
  - Strict **real-time constraints** (low jitter) for sensor data simulation.
  - Replace external hardware (sensors, haptics) with reliable software simulation.
  - Provide high-performance visual feedback using the **GLCDC** and μT-Kernel 3.0.
- **Core Opportunity**: Demonstrate **maximum utilization** of the RA8D1's Arm Cortex-M85 core in a hard real-time environment.

- **Core Idea**: Simulate the entire data pipeline using three concurrent, priority-driven tasks managed by **μT-Kernel 3.0**.
- **Key Features**:
  - Three **priority-driven tasks** (Acquisition, Processing, Visuals).
  - Replaces hardware acquisition with a **time-driven simulation loop**.
  - Visual feedback via **GLCDC** (replacing the external Flutter application).
- **Impact**: Creates an accessible, high-performance rehabilitation simulator running locally on the embedded system.

- **System**: Three **decoupled tasks** communicating via a shared state buffer (conceptual message queue) for non-blocking operation.
- **Task Structure**:

TaskName	Priority	Function/Role
Sensor_Acquisition	**Highest**	Generates5virtualADCvaluesatafixed50msinterval.
Data_Processing	Medium	Consumessensordata,executesgamelogic,calculatesreaction
Graphics_Update	Medium-Low	Renders visual feedback and game state using GLCDC/Dave2D

- The priority structure ensures the **Acquisition** task always meets its hard real-time deadline.

- **Microcontroller**: Renesas RA8D1 MCU (Arm Cortex-M85 Core, 480MHz).
- **Key Feature Utilization**:
  - **Cortex-M85 (480MHz)** : Enables rapid execution of the game logic and complex real-time calculations (Reaction Time).
  - **GLCDC & 2D Drawing Engine** : Provides smooth, responsive visual feedback, entirely replacing the external mobile application.
  - **µT-Kernel 3.0** : Guarantees deterministic, predictable task execution essential for medical/rehabilitation simulations.
- **Benefit**: The project demonstrates a complex, graphics-intensive application running entirely on-chip.

- **Criticality**: Sensor acquisition must be **deterministic** to accurately measure patient performance (reaction time, repetitions).
- **Solution: Preemption**:
  - The **Highest Priority** Acquisition Task is guaranteed to interrupt all other tasks (Processing, Graphics).
  - This ensures the *T***sample** = 50 **ms** loop integrity, removing jitter.
- **Decoupling**: Tasks are separated by data buffers, meaning the Acquisition Task never stalls waiting for the slower Graphics Task.
- This architecture proves the RA8D1's robust capability for **Hard Real-Time Operating** systems.

- **Goal**: Eliminate external dependencies to create a highly integrated, cost-effective solution.
- **Key Replacements**:

OriginalComponent	ReplacedBy	RA8D1FeatureUsed
Flex Sensors & External ADC	Sensor_Acquisition_Task	RTOS Timer
HapticFeedback(Glove)	VisualFeedback	GLCDC/2DDrawingEngine
ExternalFlutterApp	Graphics_Update_Task	ArmCortex-M85Core

- This approach maximizes the use of the RA8D1's on-chip resources.

- **Goal**: Code designed for seamless integration with the Renesas FSP and μT-Kernel 3.0.
- **Abstraction Layer**:
  - Use of placeholder macros (e.g., rtos_thread_delay, rtos_queue_send) to separate the **Application Logic** from the **RTOS API calls**.
- **Data Flow**: Clear definitions of sensor_data_t (input) and game_state_t (output) ensure structured and maintainable inter-task communication.
- **Benefit**: The application code is highly portable and easily transitions to the final e2 studio/FSP environment by replacing macros with actual API functions.

- **Achievement**: Successfully designed a complex, hard real-time, multi-tasking simulator, demonstrating the maximum potential of the RA8D1's architecture under μT-Kernel 3.0.
- **Future Work**:
  - Full integration with the **LVGL** or **GUIX** library to finalize the 30 FPS visual interface.
  - Implement the official μT-Kernel 3.0 message queue API calls instead of the conceptual global buffer.
  - Expand the game logic to include difficulty scaling and data logging to a simulated storage device.
- **Call to Action**: Ready for deployment and demonstration of real-time performance on the target RA8D1 EK hardware.

ThankYou!