Innovative look-up table operating system LUTOS & applications

Shyh-Biau Jiang1), Hsiao-Yi Yen1), Jian-An Chen1), Li-Wu Chen1), Yen-Ting Chen1), Chen-Kang Chiang1), Tse-Liang Yeh1), Li-Yet Liu 2)

1) Institute of Optical-Mechatronics Engineer, National Central University, Jhongli, Taiwan
2) ViewMove Technologies, Inc, Pingjhen City, Taiwan
Contains

- Why do we need RTOS
- About RTOS
- Programming Under LUTOS
- Look-up Table Scheduler
- Communication Between Tasks
- Communication between µ-controllers
- parameter setting and monitoring
- Distributed Logic control
- Start of Task Sequence
- Application : A Sounding Rocket Payload
Why do we need RTOS

- **Tasks of the systems**
  - Synchronous tasks: Data acquisition, filtering, feedback control
  - Asynchronous tasks: serial communication, packing, operation, setting

- **Characteristics of tasks**
  - Multi-Thread: Simultaneous execution of several tasks
  - Repeat: Regrouping Similar tasks
About RTOS

- Multi thread OS:
  - Segment of Thread: pause <=> execution
  - Sharing comp.: Multiple pause, one execution

- Generic OS vs. RTOS
  - Generic: fair sharing of comp. resources
  - Real Time: All threads completed on time

- Preemptive vs. Cooperative
  - Preemptive: snatch and segment when event happen
  - Cooperative: segment beforehand, yield after finished
Programming under LUTOS

- Task sequence: Functional block diagram
  - Task: Funct. Block, data Structure, Initialization function, Stepping Function
  - Task sequences: Connected Func. Blocks

- Recursive/Switch Case:
  - State Space: recursive update of state variable
  - Finite state machine: Switch Case & recursive update of status of state machine

- Timing Chart
  - Start of task sequence: Period and Phase
# Programming under LUTOS

## three layer structure of µC system

<table>
<thead>
<tr>
<th>Application Software</th>
<th>Operation System &amp; Driving Firmware</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Builder</td>
<td>Setting Command processor</td>
<td>CPU Execution Time</td>
</tr>
<tr>
<td></td>
<td>System State Machine</td>
<td>DATA RAM</td>
</tr>
<tr>
<td></td>
<td>Digital Filter &amp; Controller</td>
<td>Flash ROM</td>
</tr>
<tr>
<td>Task IDs Net</td>
<td>Task I/O Pointer &amp; Buffer IDs Net</td>
<td>UART</td>
</tr>
<tr>
<td>Scheduler &amp; Executor</td>
<td>Buffers</td>
<td>CAN</td>
</tr>
<tr>
<td>Flash Buffer</td>
<td>B2BBUS</td>
<td>Timer &amp; I/O ports</td>
</tr>
<tr>
<td>B2BBUS</td>
<td>CANB2B</td>
<td></td>
</tr>
<tr>
<td>Timer &amp; I/O ports</td>
<td>Sheared Timer</td>
<td></td>
</tr>
</tbody>
</table>
Look-up Table Scheduler

● Recursive Program
  ◦ State Space & Machine: One Update/iteration
  ◦ Short Execution Time: Cooperative multi-t.
  ◦ Large No. of Tasks: Low overhead needed

● Scheduling & Execution
  ◦ Task ID: One ID/task, One Trigger Flag/task
  ◦ Request for exec.: Raise Flag of task to be ex
  ◦ LUT Flag Check & Queue: Put all task with raised flag into wait for execution queue
  ◦ Execute: Get first task in queue and execute
Communication between Tasks

- **Real Time lever of Tasks:**
  - Soft RT: ASAP, Interrupt ISR
  - Non RT: Without Deadline, Low Priority Q.

- **Mixed RT levers of Task Sequence:**
  - Degrading of RT: Head Task SoftRT, Others HRT/NRT, Up-stream higher or equal to DS
  - U/D in Same lever: Don't need buffer
  - Degraded D..: Buffer data between tasks
  - Buffers: FIFO, CIRCULAR, BATCH
Communication between \( \mu \)-controllers

- **Features:**
  - Buffer2Buffer: Master \( \mu \)C, Slave \( \mu \)C, correlated Buffers
  - Physical Layer: UART, CAN
  - Commands: Read Slave no., Write Slave no. Broadcast.
  - Automatic actions: Split, reorganization, check, resend, timeout.
  - Multi-Tasking: Multiple, SRT, HRT, NRT
  - Task Trigger: Raise flag of owner task when complete, enable/disable
parameter setting and monitoring

- **Features:**
  - Communication: B2B, command buffer
  - Adjustable Parameters: Every variables of Every tasks can be open to be adjustable

- **Capabilities:**
  - Single Parameter setting and monitoring
  - Save all parameters: write flash p-page
  - Reset all parameters: read flash p-page
  - Copy Parameters: read/write p-page
  - Program update: programming/ listening mode
Distributed Logic control

- Features:
  - 4 Logic layers: Tasks, Task Sequences, Single CPU, CPU net
  - Tasks: Bottom layer, I/O Flags, status Flags.
  - Task Sequences over tasks: In head tasks, Switch Case, Sequence status, Direct read write Task data structure status and I/O flags.
  - CUP over T.S.: System State Machine, Sequence control flags, expected state of T.S. Response flags, true state of T.S.
  - CPU net: subsystem control flags, expected state, subsystem response flags, true state.
Distributed Logic control

Upper system

UCSR Buffer

System State Machine

RUCS Buffer

Sub-system 1

Sub-system 1

Sub-system 1
Start of Task Sequence

- Feature:
  - Start Task: Triggered by ISR
  - Time Based: Timer Interrupt, common multiple traffic jam
  - Shear Timer: adjustable period and phase, avoid start at same phase, limited for long period
  - Event based: Serial port, IO interrupt, Synchronization by periodic polling, TB Reserved execution time for EB
Application

- A Sounding Rocket Payload

[Diagram showing the connections between Tri. Band Ant., GPS Receiver, Main Controller, Cold Gas, RF2.4G Transmitter, Magnetometer, Gyro & GravitySensor, RS422, RS 485 Developing Bus, CAN communication Bus, Dev. Conn, DeBug Conn]
Thank You for your attention