Real-Time Multi-Modal Implementation of a Robotic Toy Car

Masud Ahmed Camille Williams Bo Peng Nathan Fisher 
{masud,ec9692,et7889,fishern}@wayne.edu. 
Wayne State University, USA.

ABSTRACT

Multiple operating modes can be found in many embedded devices (e.g., smart-phones) to efficiently utilize device resources. As similar advantages are also preferred under real-time settings, there have been researches to develop real-time multi-modal systems. However, many commercially-available embedded devices still depend on unimodal operations even for inherent multi-modal systems (e.g., automotive control system) without considering efficiency. This research implements a multi-modal system upon a low-end microcontroller to develop a robotic toy car. The goal is to reclaim additional resources through temporal isolation which may be utilized for servicing low priority applications.

1. INTRODUCTION

Multi-modal systems (MMS) are common in many systems in the form of different power savings and operational modes. These modes can be implemented in both software (SW) and hardware (HW). While SW modes enable a device to handle different tasks with varying requirements over time, hardware modes are essential for ensuring temporal isolation among applications co-executing upon a shared platform. There have been recent results that couple SW modes to HW modes in order to reap the total benefit of MMS, and developed schedulability analysis of such systems for guaranteed services even in the presence of frequent mode changes [2, 5]. However, commercially-available multi-modal real-time operations have typically been implemented in a dedicated processing environment.

The approach of a dedicated microprocessor may be suitable for small systems, but could add substantial connection/wiring overhead if multiple small subsystems constitute a larger system (e.g., cruise control in automotive system). Using temporal isolation, multiple small subsystems can be ported upon a single device to reduce the cost/size of the system. Without temporal-isolation, guaranteed service may not be possible; however, temporal-isolation may result non-continuous resources which can be modeled by HW modes. Recent research [2] addresses schedulability analysis for these systems. However, a real implementation of this approach is still pending due to the lack of support of changing modes upon non-continuous resources by real-time OSes [1, 4]. In this research, we develop a Multi-Modal Real-Time Operating System (MMRTOS) with the provision of hardware modes. As a case study, we implement a robotic car upon MMRTOS.

2. HARDWARE PLATFORM

Our MMRTOS (Figure 1) implements a multi-modal system [2] using the scheduler by Kolesnikov [3]. The target platform is an evaluation board (Arduino UNO) based on ATmega328. The board has 14 digital input/output, 6 analog inputs, and a 16 MHz crystal oscillator. For sensors/actuators, we have used four ultrasonic sensors, a servo motor, a multiplexer, and a h-bridge. Our robotic toy car has been built on top of LEGO NXT bricks and interactive servo motors. Sensors and actuators are controlled by real-time periodic tasks. The multi-modal robotic car has six running modes where each mode selects predefined set of tasks. Tasks for the servo motors have the requirements 16 µs with inter-arrival separation 300 ms. Ultrasonic sensors have the execution requirements of 80 µs with period 2.5 ms. Using temporal isolation, processor idle time is reclaimed for a low priority control task with the front ultrasonic sensor.

![Figure 1: OS for multi-modal system.](image)

Acknowledgments

This research is supported in part by an NSF Award (CNS-0953585, CNS-1116787) and an NSF REU Supplement.

3. REFERENCES