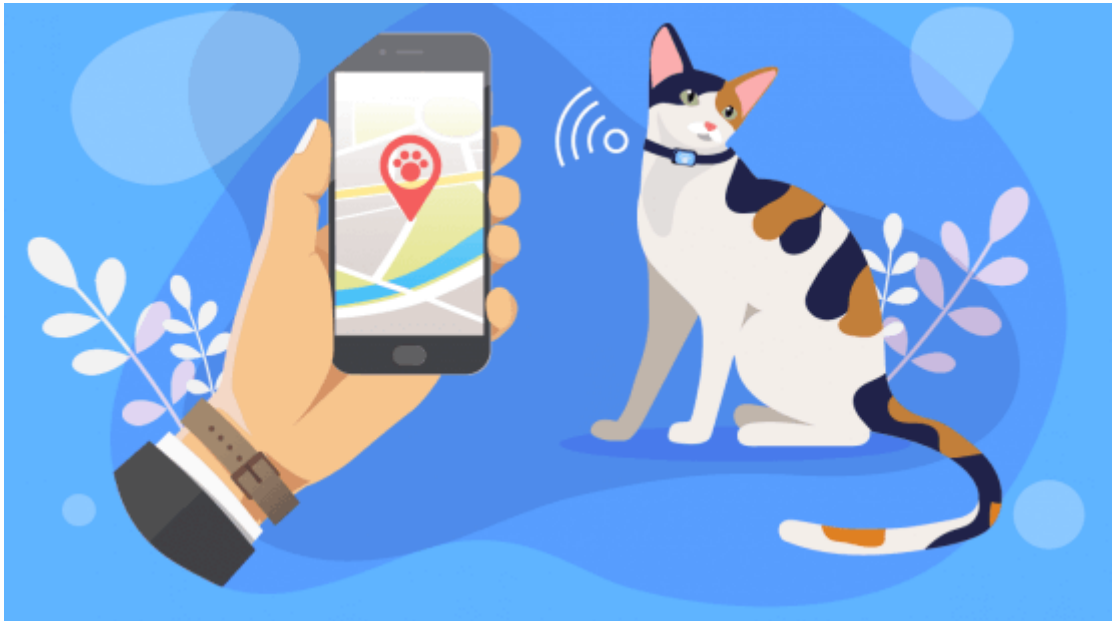


Categoría 1: Sistema de Confinamiento Virtual para Mascotas

Tutor: Javier Schandy | Co-Tutora: Andrea Delbuggio

Cantidad de proyectos: 3



El proyecto consiste en el diseño y desarrollo de un sistema embebido para el confinamiento virtual de mascotas al aire libre, utilizando un launchpad MSP-EXP430G2ET y un módulo GPS. El objetivo del sistema es permitir a los dueños de mascotas conocer la ubicación de sus mascotas en tiempo real, y recibir alertas en caso de que se alejen del perímetro de confinamiento preestablecido.

El sistema operará en dos modos: activo y pasivo. En el modo pasivo, el sistema tomará medidas de posición con una frecuencia baja y verificará que la posición esté dentro del perímetro de confinamiento preestablecido. Si el dispositivo sale del perímetro de confinamiento, entrará en modo activo, en el cual tomará medidas de posición con la máxima frecuencia y emitirá una alerta al dueño.

Una vez que el dispositivo salga de la zona de alerta, el sistema será capaz de detectar si se está alejando del perímetro o si está regresando al perímetro, generando distintos tipos de alerta en consecuencia.

Además, el sistema guardará un log de posiciones en memoria flash, lo que permitirá que el dueño de la mascota descargue y visualice la ruta que ha seguido en un mapa.

Un aspecto importante del proyecto será minimizar el consumo de energía del microcontrolador durante el funcionamiento en modo pasivo. Para lograr esto, se utilizarán técnicas de optimización de consumo y se configurará el GPS para adquirir medidas con la menor frecuencia posible en este modo.

Quedará fuera del alcance del proyecto la comunicación de los datos hacia afuera del sistema. La interacción con el dispositivo será a través de comunicación UART con un PC, los LEDs del launchpad, y opcionalmente un buzzer. El sistema sólo funcionará al aire libre y con buena señal de GPS.

Categoría 2: Análisis de Técnicas de Tiempo Real

Tutor: Julián Oreggioni | Co-Tutora: Rosina d'Eboli

Cantidad de proyectos: 3

Los sistemas embebidos de tiempo real son sistemas electrónicos especializados diseñados para cumplir una tarea específica en un plazo de tiempo determinado. Estos sistemas están destinados a interactuar con el mundo físico y, generalmente, se encuentran integrados en otros dispositivos, como vehículos, electrodomésticos, teléfonos móviles, sistemas de seguridad, equipos médicos o sistemas de control industrial. Este tipo de sistemas son esenciales en la sociedad actual.

Muchas veces confundimos el sistema "on line" con "tiempo real". Un sistema "on line" es aquel que se encuentra conectado a una red o a Internet y es accesible para ser utilizado en cualquier momento, desde otros lugares. Por otro lado, un sistema de tiempo real es aquel que debe responder y actuar ante eventos y situaciones específicas que suceden en el mundo físico, y que tiene un plazo máximo de respuesta para evitar consecuencias negativas.

Esta categoría de proyectos busca que los estudiantes experimenten con los problemas típicos que se presentan en un sistema embebido de tiempo real utilizando aplicaciones de complejidad media-baja.

Objetivos y problema a resolver

El objetivo de estos proyectos será tomar una aplicación de tiempo real, con deadlines definidos, y estudiar métodos para calcular/estimar teóricamente y medir experimentalmente los tiempos de respuesta. Evaluar si se cumplen los deadlines, y el margen que existe para modificarlos. Evaluar cómo impacta el uso de diferentes arquitecturas de software embebido en los tiempos de respuesta. Detectar cuellos de botellas ante sobrecargas, y eventualmente proponer soluciones. También se buscará evaluar el impacto en consumo de todas estas alternativas.

Posibles aplicaciones: datalogger de temperatura o datalogger de GPS (*). Ambos datalogger serán muy básicos, una implementación directa de lo realizado en el laboratorio. No es la idea que la complejidad del proyecto esté en los pormenores del desarrollo de un buen datalogger. El que implemente los drivers con el GPS no deberá llegar tan a fondo como el que trabaje con temperatura..

Actividades

1) Comparar RR c/int y encolado de funciones. Medir tiempo de respuesta con el AD2 generando salidas digitales que se activarían con la ejecución de las ISR y con los handlers. La versión básica debería arrojar que la mayor parte del tiempo las respuestas de los handlers vendrán inmediatamente después de las ISR.

2) Complejizar un poco el análisis anterior mediante:

- tener handlers que tardan más, ya sea mediante algún procesamiento de datos (filtros?) o un simple delay controlado;
- aumentar controladamente la cantidad de datos de entrada;
- aumentar controladamente la cadencia de los eventos. Puede haber eventos disparados internamente por timers (frecuencia de muestreo por ejemplo). También puede pensarse en eventos externos mediante otro micro que genere los mensajes UART o señales de E/S.

3) Lo mismo que antes pero con variantes de las arquitecturas base:

- Se podría comparar con una arquitectura que sea todo en ISR y con main vacío a ver que da. Creo que nos sorprendería que puede ser buena para algunos casos.
- Evaluar e implementar el datalogger con RR s/int. Ver hasta dónde se puede llegar por ahí.

4) Adicionales:

- Evaluar si hay herramientas (tal vez hasta CCS tenga o basada en gcc) para hacer análisis de WCET (Worst-case execution time).
- Hacer una especie de "framework" o lo que sea para agregar el manejo de los outputs (mejor sería que procese código de otros y agregue líneas) para analizar los tiempos de respuesta de "cualquier" aplicación.

Categoría 3: Reconocimiento de contenido de frecuencia de ondas sinusoidales y sincronización de movimiento con la mano

Tutor: Leonardo Barboni

Cantidad de proyectos: 3

El proyecto consiste en desarrollar un sistema para reconocimiento de frecuencia de una onda sinusoidal y copia o sincronización de la misma con el movimiento de la mano.

Funcionamiento esperado del sistema

En una de las entradas del microcontrolador deberá ir conectada una señal del AD2 por donde ingresara una onda sinusoidal de baja frecuencia (no más de 1Hz) y con un valor de amplitud como se indica abajo según el grupo.

Por otro lado, con la mano y usando sensores de aceleración se debe imitar o sincronizar el movimiento con la onda sinusoidal con la mano.

Se debe utilizar el microcontrolador para hacer el procesamiento de datos provenientes de la IMU (Inertial Measurement Unit) de la placa BOOSTXL-SENSORS. Se realizará además optimización de código por velocidad de ejecución y consumo y se debe trabajar con librerías de punto flotante. Las rutinas de atención de los sensores deberán ser **NO bloqueantes**.

La placa de sensores debe estar solidaria a la mano.

Grupo 1: copiar o sincronizar con la mano una onda sinusoidal de frecuencia 1 Hz y amplitud 2V

Grupo 2: copiar o sincronizar con la mano una onda sinusoidal de frecuencia 0,5 Hz y amplitud 1V

Grupo 3: copiar o sincronizar con la mano una onda sinusoidal de frecuencia 1 Hz y amplitud 0,5V

Una vez que se logre sincronizar con la onda sinusoidal, se debe enviar un aviso por una terminal de comandos UART al PC de que se logró la sincronización.

Categoría 4: Horno microondas.

Tutor: Leonardo Steinfeld | Co-tutora: Rosina D'Eboli

Cantidad de proyectos: 2

El proyecto consiste en modelar un horno a microondas utilizando statechart de UML e implementar el software mostrando su funcionamiento mediante una maqueta.

Las statecharts son una extensión de las máquinas de estado (vistas en el curso Diseño lógico, ya sean de Moore o Mealy) que incluyen varios agregados y mejoras en su diseño y funcionalidad. Algunas de las principales características adicionales de las statecharts incluyen: jerarquía, concurrencia y comunicación.

Se deberán considerar al menos: horno (potencia programable entre 10 y 100 en pasos de 10 emulada por ciclo de trabajo variable), grill (en principio potencia única), luz interior, teclado. Se recomienda tomar como base un microondas específico real.

Para su desarrollo se deberá implementar una cola de eventos en la que los periféricos encolen eventos para su posterior procesado. En caso de comunicación entre estados también se encolarán eventos.

Además, será posible generar eventos a través de la comunicación UART para facilitar su verificación, simulando señales de entrada (pulsadores, etc.), imprimiendo en consola las transiciones y los estados, y eventualmente las salidas.

Para la maqueta se podrá usar un display y para el teclado un arreglo de botones. En caso de no poder disponer de la totalidad de botones físicos, se podrán emular los restantes con la UART.

Antecedentes:

* Control Programable de Persiana (Smart Blind) <https://eva.fing.edu.uy/mod/page/view.php?id=31445#2022>

Referencias

* https://en.wikipedia.org/wiki/UML_state_machine

* <https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/>

Herramientas

* <https://www.smartdraw.com/state-diagram/>

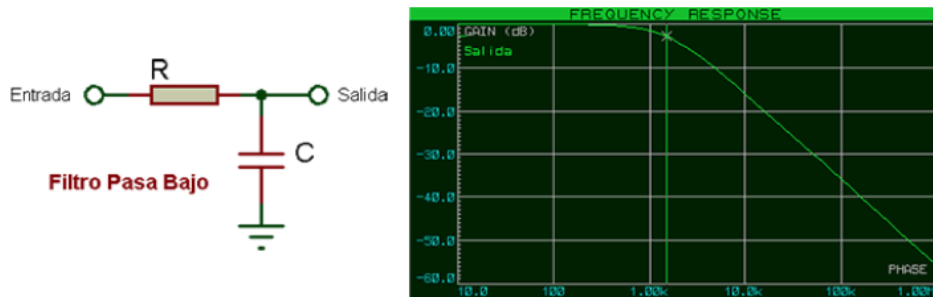
* <https://drawio-app.com/uml-state-diagrams-with-draw-io/>

Categoría 5: Sistema de Control

Tutor: Mauricio González | Co-tutor: Rodrigo García

Cantidad de proyectos: 3

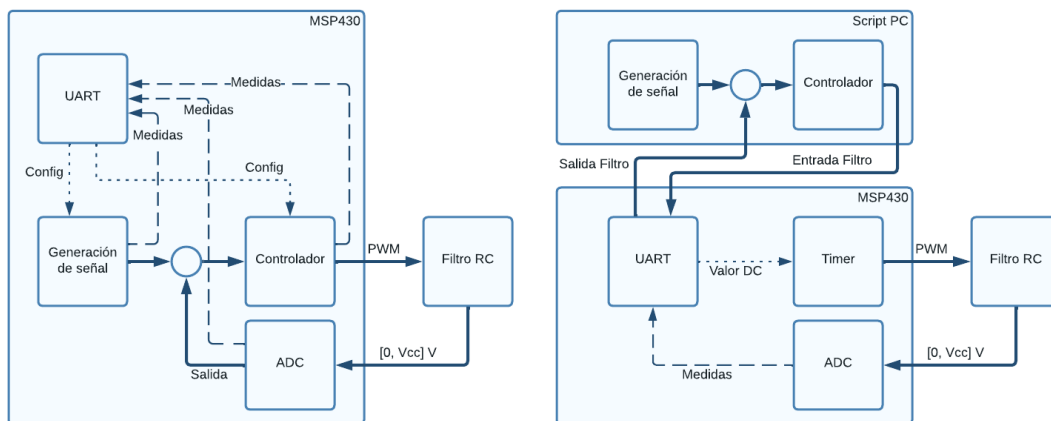
En el marco de una eventual modificación del Laboratorio de Sistemas y Control, se solicita el desarrollo de una plataforma que permita experimentar el modelado, diseño e implementación de un sistema físico y su controlador. El sistema físico sobre el que se trabajará es un sistema de una entrada y una salida, correspondiente a un filtro R-C.



El microcontrolador alimentará el circuito R-C con una señal PWM de frecuencia mucho mayor a la frecuencia de corte del filtro. Dado que el sistema sobre el que se trabaja es un pasabajos natural, la señal a la entrada del mismo será equivalente a un voltaje en continua.

Al diseñar el sistema se deberá optar por una de dos topologías:

1. Implementar el lazo de control en el microcontrolador. En esta topología, el microcontrolador actuará como generador de señales, controlador y sistema a controlar
2. Implementar el lazo de control en una computadora. En esta topología, el microcontrolador actuará solamente como sistema a controlar mientras que la PC implementará el lazo de control y la generación de señales



Algunos de los puntos a trabajar son:

- Implementar un generador PWM a partir de uno de los timers del MCU
- Implementar diferentes lazos de control discreto en el MCU. Se podrá evaluar la implementación de transferencias discretas arbitrarias o, más sencillamente, controladores PID
- Implementar un mecanismo de anti-windup
- Implementar un protocolo de comunicación con el MCU que permita
 - Reportar medidas de la entrada y la salida del filtro a la PC
 - Transferir parámetros del controlador al MCU
 - Transferir una señal de entrada al MCU
 - Iniciar/Detener un experimento

Es deseable que los estudiantes que seleccionen esta categoría tengan un dominio general sobre teoría de sistemas y control (funciones de transferencia, estabilidad, controladores PID) pero no es excluyente.