Categoría 1: Compresión de datos para Sistemas Embebidos

Tutor: Javier Schandy
Cantidad de proyectos: 2

Descripción:

Este proyecto se centra en la implementación y comparación de algoritmos de compresión de datos en el contexto de una aplicación de adquisición y transmisión de datos en tiempo real.

La compresión de datos en sistemas embebidos tiene una importancia crítica debido a sus limitaciones inherentes en términos de memoria, capacidad de procesamiento y consumo energético. Comprimir los datos permite mejorar la eficiencia para el almacenamiento y transmisión de datos. Aplicaciones típicas que se benefician de la compresión de datos son las de adquisición de datos de muy alta frecuencia, o los procesos de actualización de firmware, en donde se transmiten y almacenan grandes cantidades de datos.

Para la compresión, en principio se sugiere evaluar los algoritmos Huffman Coding y Run-Length Encoding (RLE), pero se podrán considerar otros. En principio el único hardware necesario para realizar el proyecto será el LaunchPad MSP430G2, y se implementará un sistema funcional de adquisición, almacenamiento, compresión y transmisión de datos.

Tareas:

- **Adquisición:** Implementar adquisición de datos utilizando el ADC del microcontrolador a una frecuencia a configurar. En principio se configurarán frecuencias de adquisición del órden de las decenas o centenas de Hertz. Se pueden tomar datos del sensor de temperatura interno del microcontrolador, o de algún sensor analógico conectado a los pines del launchpad.
- **Almacenamiento y compresión de los datos:** Inmediatamente después de la adquisición, los datos deberán ser almacenados y comprimidos utilizando los algoritmos como Huffman Coding y Run-Length Encoding (RLE). Se enfatizará la importancia de la eficiencia de la compresión, tanto en términos de reducción del tamaño de los datos como en el uso de recursos del sistema.
- **Transmisión de datos comprimidos:** Los datos comprimidos se enviarán a través de UART a una PC para su análisis y visualización.
- **Evaluación y Comparación:** Los algoritmos implementados serán evaluados y comparados en base a los siguientes criterios:
    - **Rendimiento de compresión**: Eficiencia en la reducción del tamaño de los datos.
    - **Tiempo de ejecución**: Velocidad de la compresión y descompresión de datos.
    - **Consumo energético**.
    - **Footprint/Uso de memoria**: Cuánta memoria utiliza el algoritmo.

Categoría 2: Mejora de precisión en Sistemas Globales de Navegación por Satélite

Tutor: Julián Oreggioni
Cantidad de proyectos: 2

Los Sistemas Globales de Navegación por Satélite (GNSS por sus siglas en inglés), generalmente incorrectamente referidos en forma genérica como GPS[1], refieren al conjunto de tecnologías de posicionamiento satelital empleadas para determinar rápidamente la ubicación geográfica de un objeto.

Los receptores estándar de GNSS de bajo costo tienen una precisión en la ubicación que puede ser superior a los 10 metros. Existen varias técnicas para mejorar esa precisión que tienen diferentes características: DGPS (GPS diferencial) estático, DGPS/RTK (Real Time Kinematic DGPS), AGPS (GPS asistido), usando información de radiobases en caso de tenerla disponible, entre otras.

El Inst. Geográfico Militar de Uruguay, tiene una estación DGPS/RTK en Montevideo y da acceso a este servicio en tiempo real y gratuito.

El objetivo general del proyecto es introducirse en la temática de los GNSS, entender su funcionamiento básico, las fuentes de error, e implementar algún mecanismo sencillo para mejorar la precisión.

Ejemplos de posibles proyectos:

1) Implementar una estación DGPS estático. Se trata de un sistema embebido que se ubica al aire libre y está fijo en un punto con coordenadas conocidas (calibrado con 12 cm de incertidumbre en el estacionamiento SUR del Fing), calcula los datos de corrección para DGPS estático, los muestra en un display OLED y los envía por UART a una Laptop, donde pueden ser recibidos, procesados y transmitidos. Idealmente se muestran los datos en un mapa en la Laptop.

2) Implementar un nodo DGPS estático: Se trata de un sistema embebido que se ubica al aire libre y que se mueve libremente alrededor de la Estación DGPS estático, recibe datos de ubicación desde un Receptor de GNSS, recibe datos para corrección desde una Laptop (UART), corrige la ubicación y muestra en un display OLED las coordenadas de los datos originales y datos corregidos. Idealmente se muestran todos los datos en un mapa en la Laptop.

3) Implementar la Estación DGPS estático y 1 nodo DGPS estático (sin display OLED).

4) Implementar algún otro mecanismo de corrección, como por ejemplo DGPS/RTK, AGPS u otros. Para usar RTK, hay que terminar de confirmar si el receptor GNSS sirve. En ese caso, se trata de un sistema embebido que se ubica al aire libre y que se mueve libremente, recibe datos de ubicación desde un Receptor de GNSS/RTK, recibe datos para corrección desde una Laptop por UART (que a su vez los recibió desde la estación del Inst. Geográfico Militar de Uruguay en Montevideo), corrige la ubicación y muestra en un display las coordenadas de los datos originales y datos corregidos. No hay display OLED. Idealmente se muestran todos los datos en un mapa en la Laptop.

Hardware a utilizar

[1] GPS (Sistema de Posicionamiento Global) es un sistema lanzado y operado por los Estados Unidos de América. Existen otros, como GLONASS de la Federación Rusa, GALILEO desarrollado por la Unión Europea, y BDS de China, entre otros.

- Receptor GNSS modelo Neo 6M de Ublox (https://www.u-blox.com/en/product/neo-6-series)
- Launchpad MSP-EXP430G2ET: tiene una única UART, por lo cual deberá implementarse una UART por software.
- Display OLED
- AD2

Antecedentes a estudiar

- GNSS y DGPS
- Servicio brindado por el Inst. Geográfico Militar: https://igm.gub.uy/2016/05/20/servicios-regna-rou/

Proyectos sisem de referencia:

- 2023: Análisis de técnicas de tiempo real en un sistema de adquisición y transmisión de ubicación geográfica (GPS-TReal). Proyecto que desarrolló driver de UART por software, y driver de Receptor GNSS modelo Neo 6M de Ublox
- 2022: Heart Rate and Oxygen Saturation meter (HeroxS). Proyecto con manejo de display OLED.

Herramientas adicionales a las del curso:

- Software UBLOX Center
- Buen manejo de Python puede ser útil, aunque no es excluyente

Categoría 3: Embedded Machine Learning

Tutor: Leonardo Barboni
Cantidad de proyectos: 2

El aprendizaje automático y procesamiento de señal diminuto (TinyML) es una nueva frontera del aprendizaje automático y sistemas embebidos. Al integrar modelos de aprendizaje automático y procesamiento de señal en dispositivos y microcontroladores (por ejemplo en MCU para Internet de las Cosas), ampliamos el alcance de las aplicaciones y permitimos una inteligencia ubicua. Sin embargo, TinyML es un desafío debido a las limitaciones que impone el hardware como los pequeños recursos de memoria y cálculo que están disponibles en los microcontroladores. Por eso es importante co-diseñar los algoritmos en función de las limitaciones del hardware de la aplicación..

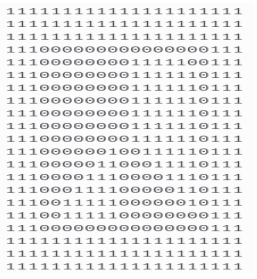
Por otro lado, la detección de bordes en imágenes es una de las tareas fundamentales de la "visión por computadora". Su objetivo principal es extraer el límite del objeto y el borde que contiene la información principal de la imagen, ignorando otros detalles con menor importancia. El borde y el límite de los objetos son muy importantes en aplicaciones de TinyML.

Esta categoría de proyectos tiene como objetivo experimentar con los problemas que se presentan en la implementación de algoritmos de complejidad media-baja y manejo de grandes volúmenes de datos (relativos a la memoria del MCU) en un sistema embebido de tiempo real.

Se propone trabajar con la imagen que se muestra en la Figura 1 (a) y con una representación en 0-1 como se muestra en Figura 1 (b). El tamaño de la imagen es de 21 x 21 pixels = 441 bytes (cada pixel es un byte y recordar que el MSP430G2553 tiene 512 bytes de SRAM). Esta imagen es entregada al grupo de estudiantes en un archivo de texto **[1]** y debe cargarse en el Launchpad para detectar sus bordes.

La nueva imagen generada solo con los bordes debe pasarse a un PC mediante UART para mostrar el resultado. Se debe implementar sobre la imagen las siguientes operaciones: ***a)*** un algoritmo fuzzy de detección de bordes basado en las 8 reglas desarrolladas en **[2]** y ***b)*** un clásico operador Sobel **[3]**

(a)



(b)

Figura 1: (a) Imagen que se utilizará en este proyecto, (b) representación de la figura en un mapa de valores 0-1 para utilizar en el MSP-EXP430G2ET con un microcontrolador MSP430G2553.

Algunas tareas que se deberán cumplir obligatoriamente:

1- Incorporar la imagen interpretada como un array en el programa, implementar los dos algoritmos y aplicarlo sobre la imagen dada y sacar la imagen resultante al PC mediante UART

2- Presentar en la documentación las medidas de tiempo de ejecución de los algoritmos (la medida se realizará mediante un timer que debe ser iniciado y finalizado sincronizado con la ejecución del algoritmo)
3- Explorar y presentar en la documentación todo el uso de memoria y su mapa (donde empieza, termina y cuanto ocupa cada dato, función y variable). Decir cuanta memoria queda disponible en los programas finales.
4- En algún punto del caso algoritmo Sobel **[3]**, se sugiere utilizar en C la expresión :
magnitude = sqrt(pow(Gx, 2) + pow(Gy, 2)); (pero hay alternativas a su uso en el caso de este proyecto). Implementar la expresión anterior así como también una alternativa que la aproxime. Decir cuánto ocupan de memoria y cuanto tiempo demoran en su ejecución.
5- Mostrar medidas de consumo con Energy Trace
6- Analizar las sugerencias del Ultra Low Power (ULP) Advisor que aparecen en la consola al compilar e implemente los consejos dados. ¿Se obtiene algún cambio al usar las sugerencias?

Referencias y Enlaces

[1] Archivo de texto que contiene un array de 441 bytes llamado *geometria_binaria_array21x21.txt* https://drive.google.com/file/d/1I6rFx8itl4iFW23yQmy2Ct5PUt1i4XHk/view?usp=sharing

[2] A.k. Pandey, H.R.S.S.N. Chatla, M. Pandya, A. Farhan, A. S. Rana,"Image Edge Detection Using Fuzzy Logic Controller" Published in: 2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON) DOI: 10.1109/REEDCON57544.2023.10150762

[3] Sobre operadores Sobel para detección de bordes
https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm
https://automaticaddison.com/how-the-sobel-operator-works/

Categoría 4: Emulador de receptor GNSS

Tutor: Mauricio Gonzalez
Cantidad de proyectos: 2

Un receptor GNSS (Sistema Global de Navegación por Satélite) es un dispositivo que interpreta las señales de los satélites de geolocalización para determinar la ubicación geográfica precisa de un objeto en la Tierra. Muchos de los receptores GNSS disponibles para su integración en sistemas embebidos consisten en módulos con comunicación UART capaces de comunicarse con otros sistemas mediante comandos en formato NMEA.

La complejidad de los comandos NMEA hace que muchas veces la configuración y utilización de los receptores GNSS sea laboriosa. Adicionalmente, la necesidad de utilizar estos equipos a cielo abierto agrega una capa de dificultad adicional.

El objetivo de este proyecto es el desarrollo de un simulador de receptor NMEA basado en un Launchpad de MSP430. El modelo de receptor GNSS a simular será el Neo 6M de Ublox.

Tareas a desarrollar

- Sniffear la comunicación UART entre un Neo 6M y el software UBlox Center
- Interpretar los mensajes recolectados
- Estudiar la documentación del Ublox Neo 6M, definir los comandos soportados por el simulador
- Implementar un handler de comandos NMEA
- Implementar un módulo que controle el comportamiento del simulador en base a comandos enviados por una interfaz a definir

Hardware a utilizar

- Receptor GNSS modelo Neo 6M de Ublox (https://www.u-blox.com/en/product/neo-6-series)
- Launchpad MSP-EXP430G2ET: tiene una única UART, por lo cual deberá implementarse una UART por software.
- Analizador lógico del AD”

Categoría 5: Control infrarrojo multi-protocolo

Tutor: Rodrigo García
Cantidad de proyectos: 2

El objetivo principal es desarrollar un control remoto para dispositivos como televisores, aires acondicionados entre otros. El control remoto será programable y deberá transmitir comandos utilizando al menos dos protocolos de comunicación establecidos. Además, se busca permitir al usuario seleccionar el protocolo deseado y asignar distintos comandos a cada botón del control remoto.
Los protocolos sugeridos son:

- NEC
- SIRC
- RC-5

El sistema propuesto consta de los siguientes componentes:

- Launchpad MSP-EXP430G2ET: Se utilizará esta plataforma de desarrollo de microcontroladores como núcleo del sistema de control remoto.
- Teclado Matricial: Se empleará un teclado matricial como la interfaz de entrada para el control remoto.
- LED Infrarrojo: Se utilizará para la transmisión de los comandos desde el control remoto al dispositivo receptor.

Mediante UART se debe poder seleccionar dos modos de operación:

- Modo de programación:
  En este se podrá seleccionar el protocolo a utilizar y también al presionar un botón se entrará en la programación del mismo, mediante la comunicación serial se le indicará que comando tendrá este botón.
- Modo de uso:
  Este será el modo normal de operación donde al presionar un botón se transmita utilizando el LED el comando previamente guardado.

Antecedentes:
CONTROLin
µWave

Referencias:
NEC Protocol
Philips RC-5 Protocol
Sony SIRC Protocol

Categoría 6: Monitor de sueño
Tutoras: Josefina Lema, Varinia Cabrera
Cantidad de proyectos: 2

El proyecto consiste en desarrollar un monitor de posición durante el sueño, utilizando un microcontrolador (MSP430G2553) y datos de la IMU (Unidad de Medición Inercial) de la placa BOOSTXL-SENSORS de Texas Instruments. Una posible aplicación de esto sería ayudar a los usuarios a entender sus hábitos de sueño y potencialmente mejorar la calidad de su descanso al identificar patrones o posiciones que favorezcan un mejor descanso.

Además, el dispositivo debe incluir un gestor de energía. Dependiendo de la energía disponible, el sistema ajustará los parámetros de operación del dispositivo (sensor y del microcontrolador) para disminuir su consumo energético y así prolongar su autonomía sin comprometer su funcionamiento. Para implementar este gestor, es necesario monitorear el nivel de energía remanente midiendo la tensión de alimentación (por ejemplo, supercondensador o batería). Para probar el concepto también se podría controlar una tensión para simular diferentes niveles de energía en el sistema y evaluar el comportamiento adaptativo.

Para evaluar el funcionamiento del sistema, se definirán ciertas posiciones base (boca arriba, de lado, boca abajo) que el sistema debe reconocer durante el período de descanso. Se establecerán umbrales o criterios para detectar estas posiciones, y el sistema deberá ser capaz de transmitir a una PC las posiciones registradas durante dicho período.

Se deben implementar técnicas para minimizar el consumo obteniendo diferentes niveles de consumo-prestaciones. Por ejemplo, puede disminuir la frecuencia de muestreo del sensor y/o simplificar el procesamiento a realizar para la determinación de las diferentes posiciones para disminuir el consumo del dispositivo. Se deberá caracterizar el consumo obtenido para cada nivel y mostrar el comportamiento adaptativo del dispositivo.