

Desarrollo de un equipo para grabar los sonidos de los bovinos a la intemperie

S. Reyes, V. Olivera, M. C. San Román y J. P. Oliver

Facultad de Ingeniería, Universidad de la República

Montevideo, Uruguay

santiagoreyesgonzalez@gmail.com, valeoli@gmail.com, cecisr@gmail.com, jpo@fing.edu.uy

Resumen—En este trabajo se presenta un dispositivo capaz de grabar y almacenar el sonido que emiten los bovinos durante la masticación e ingesta de forraje. Este dispositivo, junto con un software capaz de detectar las actividades realizadas por el animal (pastoreo, rumia y descanso) a partir de las señales de audio obtenidas, forma parte del prototipo Monitoreo del Sonido Bovino (MOSOBO). A partir de estos datos es posible determinar en qué momento del día y durante cuánto tiempo el animal realizó alguna actividad, lo que permite estudiar el comportamiento ingestivo de los rumiantes.

El dispositivo tiene una autonomía energética de más de 24 horas, graba y almacena audio de alta calidad, no afecta el comportamiento del animal, es de fácil colocación, bajo costo, estable y robusto.

Debido a la buena calidad del audio adquirido y al procesamiento realizado a la señales de audio se lograron excelentes resultados al reconocer las diferentes actividades que el animal realiza (pastoreo, rumia y descanso).

Index Terms—SBC, Linux embebido

I. INTRODUCCIÓN

El objetivo de este trabajo fue desarrollar a nivel de prototipo un dispositivo capaz de grabar el sonido que realizan los bovinos durante la masticación e ingesta de forraje, junto con un paquete de software que corre en un PC, capaz de determinar y desplegar qué actividades realizó el animal (pastoreo, rumia o descanso), en qué orden y la duración de las mismas.

El comportamiento animal en pastoreo es resultado de factores abióticos (distancia al agua, la temperatura, la luz, el pH, el suelo, nutrientes, pendiente) y factores bióticos (cantidad y calidad del forraje por animal, estado fisiológico y metabólico del animal)[1]. Justamente la cantidad de forraje consumida por el animal depende del *tiempo de pastoreo* y su tasa de consumo[2].

Existen situaciones en las que el consumo de energía del animal no se modifica pero su productividad se ve modificada como consecuencia de tener diferente comportamiento ingestivo. La medición del comportamiento y del consumo de energía contribuiría a explicar gran parte de los resultados en productividad animal a nivel experimental y comercial.

Los requisitos principales del dispositivo consistieron en que el sistema fuera robusto (ya que se utiliza a la intemperie junto con el animal), que tuviera autonomía energética por más de 24 horas, fuera económico, de fácil colocación y bajo peso, y que fuera inofensivo y no modificara el comportamiento de los animales.

Con respecto al software el requisito principal fue el de ser capaz de identificar 3 actividades bien diferenciadas (pastoreo, rumia y descanso), en qué momento son realizadas y la duración de las mismas.

Este trabajo se basó en las conclusiones de algunas investigaciones sobre el comportamiento de los rumiantes y los sonidos que los mismos emiten en las distintas etapas de la digestión[3].

En las siguientes secciones se describen los componentes seleccionados para realizar el dispositivo y como se implementó el software del sistema embebido. Finalmente se mencionan las conclusiones y perspectivas de este trabajo.

II. HARDWARE

Para poder determinar los requerimientos de frecuencia de muestreo y cantidad de bits del dispositivo de grabación fue necesario conocer qué características tenían los sonidos emitidos por los bovinos. De la buena elección de estos requerimientos dependía el éxito del software de análisis y reconocimiento.

Por esta razón, primero se realizaron varios ensayos para obtener señales de audio de los sonidos emitidos por los bovinos. Luego de procesar dichas señales, se determinaron los requerimientos del dispositivo y se seleccionaron los componentes.

Ensayos preliminares

Para obtener las señales de audio, se seleccionaron varios componentes con distintas características y se realizaron varios ensayos en 2 vacas distintas (Ver Figura 1).

Procesamiento digital de las señales

Para analizar las señales obtenidas a partir de los ensayos, se utilizó el software *Wavesurfer*. Con el mismo se observaron y obtuvieron los segmentos de las señales que aportaron información útil.

En la Figura 2(a) y en la Figura 2(b) se pueden observar algunas secuencias representativas de las actividades de rumia y pastoreo respectivamente.

La Figura 2(a) no brinda información sobre la frecuencia mínima de muestreo, pues se obtuvo con un reproductor mp3 el cual muestrea a 8 kHz. Sin embargo, se puede notar fácilmente que la señal presenta un patrón rítmico.

En la Figura 2(b) se pueden identificar componentes de señal de interés de hasta 6 kHz. Esta señal a diferencia de

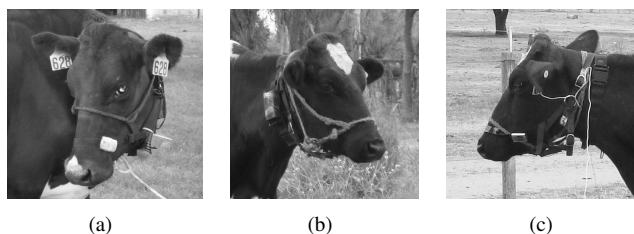
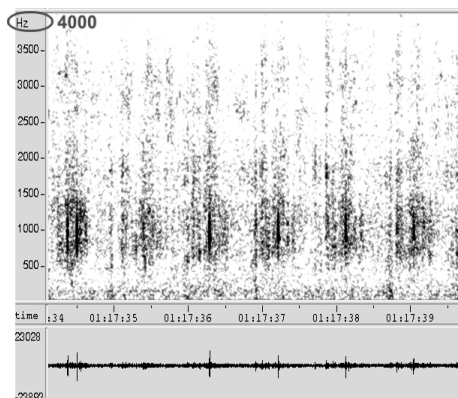
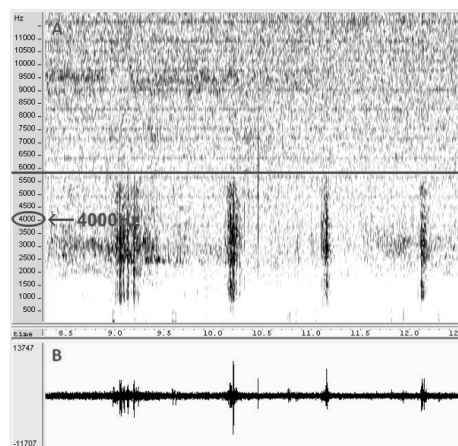


Figura 1. Ensayos realizados para adquirir las señales a analizar



(a) Señal de audio emitida por un bovino durante la rumia.



(b) Señal de audio emitida por un bovino durante el pastoreo.

Figura 2. Señales obtenidas a partir de los ensayos.

la anterior, fue obtenida con un micrófono de PC y una computadora.

Determinación de los requerimientos

El artículo “Computational method for segmentation and classification of ingestive sounds in sheep”[4] propone un método novedoso para analizar y reconocer automáticamente señales sonoras emitidas por ovinos al realizar pastoreo y

rumia¹.

En base a este artículo y a las características presentes en las señales observadas durante el procesamiento digital de las mismas se decidió que el dispositivo de grabación debía muestrear al menos a 22 kHz y tener una resolución de al menos 12 bits.

Teniendo en cuenta que el dispositivo debía ser capaz de grabar el sonido emitido por los rumiantes durante 24 horas, al obtener las señales a 22 kHz, 12 bits, fue necesario contar con una memoria no volátil de al menos 3,5 GB.

Era indispensable que el dispositivo permaneciera en el animal durante 1 día sin interrupción, por lo que la batería debía ser capaz de alimentar el dispositivo durante este tiempo. Además debe tener un factor de forma aceptable y ser liviana. Teniendo en cuenta el presupuesto con el que contaba el proyecto y la relación entre la capacidad de las baterías y su peso, concluimos que la placa debía consumir menos de 1,5 W.

Uno de los objetivos del proyecto fue determinar en qué momento del día el animal realizó cada actividad, por lo que era necesario incluir un reloj de tiempo real.

Una de las características deseables del dispositivo era la capacidad de interactuar con el usuario, para lo cual se decidió utilizar LEDs y botones.

Del análisis de los distintos requerimientos del sistema se llegó a la siguiente especificación:

- Muestreo a 22 kHz, al menos 12 bits.
- Memoria no volátil de 3,5 GB.
- Bajo consumo, menor a 1,5 W.
- Real Time Clock o posibilidad de conectar uno.
- Botones para la interfaz con el usuario (2 o 3).
- 2 LEDs.

Elección de los componentes

Se analizaron varias opciones para desarrollar el dispositivo.

La primera opción fue la de basar nuestro desarrollo utilizando grabadoras digitales de audio, reproductores de mp3, mp4 o celulares capaces de adquirir audio; pero se descartó dado que ninguno cumplía los requerimientos mínimos en frecuencia de grabación y/o cantidad de bits, y costo.

Luego se consideró la opción de desarrollar a medida placas con microcontroladores y ADCs o tarjetas de audio (o utilizar

¹El método propuesto puede ser utilizado indistintamente en bovinos y ovinos ya que desde el punto de vista del comportamiento ingestivo son considerados iguales.

algún kit de desarrollo existente). Esta opción si bien presenta ventajas en cuanto a consumo y costos, implicaba un riesgo para cumplir los plazos del proyecto y debido a que había que tomar la definición en una etapa temprana del mismo en la cual todavía no se tenían claros todos los requerimientos se decidió utilizar un sistema con mayores recursos y flexibilidad.

Por las razones antes mencionadas finalmente se decidió utilizar una *Single Board Computers* que incluyera un sistema operativo para acortar tiempos de desarrollo y una tarjeta de sonido o ADC externo que cumpliera los requerimientos. Esta solución permitiría tener mayor flexibilidad y velocidad de desarrollo de las aplicaciones, y soportar futuras ampliaciones del equipo como son transmitir los datos mediante Wi-Fi facilitando la descarga de datos del dispositivo o añadir un GPS que permitiría relacionar las actividades realizadas por el animal y la ubicación geográfica del mismo.

SBC, tarjeta de sonido y unidades de almacenamiento: Luego de evaluar las placas de la Tabla I y comprobar si cumplían los requerimientos estipulados (con pequeños agregados), concluimos que la placa TS-7260 de Technologic System era la mejor opción. Además se decidió utilizar la tarjeta de sonido USB 3D Sound modelo HY554 por cumplir con los requerimientos de frecuencia de grabación y cantidad de bits y ser de bajo costo.

TS-7260: Tiene instalado de fábrica el núcleo 2.4.26 de GNU/Linux, y está optimizado para la arquitectura de la misma y su consumo máximo se encuentra en el orden de 1 W (sin el uso de periféricos y a máxima frecuencia).

Por defecto la TS-7260 trae instalada la aplicación BusyBox[9]; pero la utilización de la misma fue descartada debido a distintas limitaciones². Se decidió instalar una distribución GNU/Linux Debian Sarge 3.1 debido a su compatibilidad con nuestra arquitectura y ser un sistema sumamente estable.

USB 3D Sound modelo HY554: Inicialmente se había decidido grabar a 22 kHz pero la tarjeta de sonido seleccionada muestrea únicamente a 48 kHz; por lo que finalmente la grabación de audio se realiza con frecuencia de muestreo de 48 kHz.

Unidades de almacenamiento: Para poder almacenar 24 hs a 48 kHz, 16 bits, se necesita una unidad de almacenamiento de al menos 7 GB. Se intentó grabar audio y decimar los archivos ya grabados, simultáneamente en la placa pero los mismos resultaban corruptos. Tampoco fue posible utilizar algoritmos de compresión de audio con pérdida de información porque no cumplíamos con los requisitos mínimos de 22 kHz, 12 bits.

La distribución GNU/Linux Debian Sarge 3.1 ocupa alrededor de 512 MB por lo que se necesita una unidad externa de almacenamiento dado que la memoria interna de la TS-7260 no es suficiente.

Finalmente se decidió almacenar los datos en un pendrive de 8 GB e iniciar el sistema desde una memoria SD card de 2 GB.

²Con la aplicación BusyBox no es posible manejar módulos de audio, interfaz de red y almacenamiento.

Batería: Para determinar la capacidad de la batería a utilizar se relevó el consumo de la placa en las posibles situaciones de funcionamiento (ver Tabla II): inicio del sistema operativo, estado de espera³ o grabando y almacenando.

Tabla II
CONSUMO DEL SISTEMA EMBEBIDO AL SER ALIMENTADO POR UNA FUENTE DE 5 V.

Actividad	Consumo (mA)	Potencia (W)
Inicio del sistema	300 máx	1,5
Estado de espera	220	1,1
Grabando y almacenando	240	1,2

Eligiendo el peor caso (inicio del sistema), el consumo fue de 1,5 W, lo cual nos llevó a consumir 36 Wh en un día.

Teniendo en cuenta la potencia que debía suministrar, el rango de alimentación de la SBC⁴, factor de forma aceptable, el costo y la relación entre su peso y la capacidad; se decidió utilizar en el primer prototipo 2 baterías de gel (modelo RB632C) de 6 V y 3200 mAh. Usando ambas baterías se llegó a una capacidad de 38,4 Wh, cumpliendo los requerimientos planteados anteriormente. Las baterías seleccionadas pesan 1200 g y tienen un factor de forma aceptable: 32 x 96 x 60 mm.

Para el diseño final se utilizará un pack de 5 baterías de litio-ion de 2800 mAh y 3.7 V (modelo GTL), con lo cual el peso de las baterías bajaría significativamente quedando en aproximadamente 200 g.

Micrófono: Se decidió utilizar el micrófono Ht-101 de la marca Xtreme que se encontraba en el mercado local a un precio accesible, ancho de banda: 100-16000 Hz, impedancia: 2.2 kΩ, sensibilidad: -55db ± 3db y era omni direccional.

Placa de interfaz de usuario y control de energía: Para poder mantener una comunicación con el usuario se diseñó una placa interfaz de usuario y control de energía. La misma cuenta con tres botones con las siguientes funcionalidades: encendido, apagado y grabación.

Uno de los problemas que se presentó al apagar la placa es que la misma sigue tomando corriente luego de que el sistema operativo es apagado. Para evitar este problema se

Como se puede observar en la Figura 3, uno de los botones mecánicos se encuentra en paralelo a un botón lógico. El botón lógico se realizó mediante un opto acoplador y un transistor los cuales se disponen en una configuración Darlington. Al presionar el botón de encendido la placa se prende y ejecuta su secuencia de arranque. Unos segundos después el sistema toma el control de los puertos de entrada-salida, se polariza el transistor Q1 poniendo en "1" lógico el pin 7 del DIO, activando así al opto acoplador.

El transistor que realiza la salida del opto acoplador funciona como un interruptor ya que queda en estado de saturación. Esto a su vez produce la saturación del transistor Q2 permitiendo así el pasaje de la corriente. El proceso de encendido tiene una duración aproximada de 4 segundos y

³Sistema luego del arranque a la espera de una orden del usuario.

⁴La TS-7260 debe ser alimentada entre 4,5 y 20 V.

Tabla I
COMPARACIÓN DE POSIBLES PLACAS

Sinlge board computers seleccionadas				
	Gumstix Overo Earth[5]	PPM-LX800[6]	EPX-GX500[7]	ARM TS-7260[8]
Procesador	ARM Cortex-A8 CPU 600 MHz	AMD 500 MHz Geode LX800	AMD Geode™ GX500	200MHz ARM9 CPU
Memoria	256 MB RAM 256 MB Flash	Up to 128 MB SDRAM Up to 64 MB of AMD MirrorBit Flash	Up to 512 MB SDRAM	32MB SDRAM 32MB NAND Flash
RTC				Si
ADC	10 bits		12-bit	2 12-bit
SD	2GB micro SD	Si	Si	Si
USB	USB HS Host	2 x 2.0 ports	2 X Two USB 1.1 ports	2 USB 2.0(12 Mbit/s max)
Tamaño (mm x mm x mm)	17 x 58 x 4.2	90 x 96		95 x 120
Consumo	1W	0.9W	1.0W	0.25W a la frecuencia mínima
Sist. Operativos que soporta		Windows XP, XP Embedded, Linux, DOS, x86 RTOS	Windows XP, XP Embedded, Linux, Windows CE, DOS, x86 RTOS	Linux out-of-the-box

cuando finaliza enciende el led verde, indicando al usuario que el sistema está encendido y que puede soltar el pulsador.

La última acción que toma el sistema operativo de la placa TS-7260 es bajar el pin 7 a "0", esto hace que la SBC se quede sin alimentación, por lo que con esta solución la placa efectivamente no consumirá cuando esté apagada.

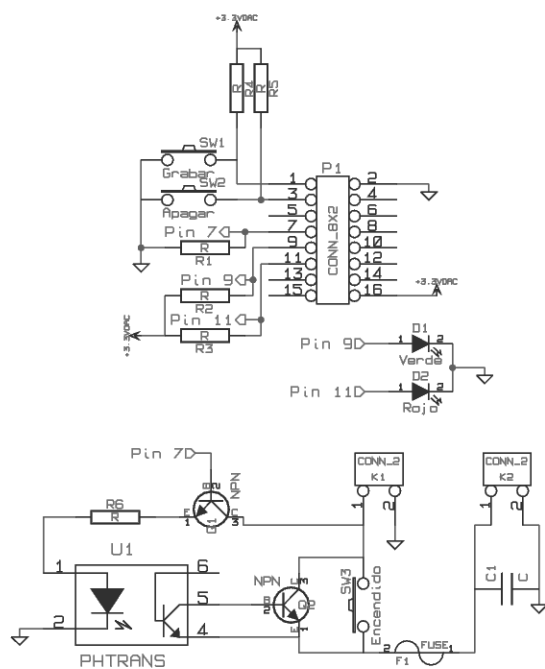


Figura 3. Esquema de la placa interfaz usuario y control de energía.

III. DISEÑO INDUSTRIAL

La caja seleccionada para contener la TS-7260, tarjeta de audio, baterías y placa interfaz es una *Pelican 1060 Micro Case*[10] es a prueba de polvo, agua y golpes (ip67); y sus dimensiones son 20.9 x 10.8 x 5.7 cm.



Figura 4. Bozal, caja con el dispositivo y micrófono.

En la Figura 4, se puede observar el bozal diseñado, en donde se colocan el micrófono y la caja con el dispositivo. Para poder conectar el micrófono a la placa de audio se colocó un pasacables estanco para mantener las propiedades de la caja.

IV. ESPECIFICACIÓN DEL SOFTWARE DEL SISTEMA EMBEBIDO

Introducción

El diseño del sistema embebido se divide en los siguientes bloques:

- Inicialización del sistema.
- Adquisición y almacenando de audio.
- Apagar el sistema.

Varias de las acciones que debe realizar el sistema embebido son realizadas por programas y/o librerías de código abierto ya implementadas y testeadas por varias comunidades de desarrolladores.

A continuación se detalla el funcionamiento de los bloques.

Inicialización del sistema

Luego de presionar el pulsador de encendido se ejecuta TS-SDBOOT e inicia el sistema operativo.

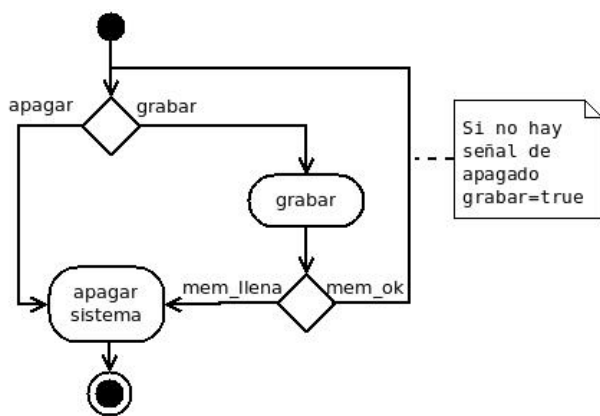


Figura 5. Interacción de los programas del software del sisem.

Ejecutar TS-SDBOOT: El TS-SDBOOT es un bootloader para las placas TS-7260, TS-7300 y TS-7400. Al ejecutarlo se levanta la imagen del kernel que se encuentra en la primera partición de la SD card para luego descomprimirla en RAM.

Luego se ejecuta linuxrc, el cual es un archivo realizado en shell scripting que se encarga de configurar el sistema de archivos del GNU/Linux.

Después que todas estas acciones han finalizado, se levanta el sistema operativo Debian Sarge 3.1, el cual se encuentra en la tercer partición de la SD card.

Programa linuxrc: Se ejecuta para poder configurar y trabajar desde la SD card como se diseñó, modificar la configuración del sistema de archivos de la tercera partición de la SD card y realizar el control de LEDs y pulsador para encender la placa.

Inicio del sistema operativo: Para que el sistema arranque automáticamente con nuestro software se modificó la autenticación de los usuarios y los demonios que se levantan al inicio del sistema.

Adquisición y almacenando de audio

La adquisición del audio se realiza mediante el uso del software Bplay, el cual cuenta con una interfaz de grabación llamada brec. Se debe configurar el software para grabar a la frecuencia de muestreo y cantidad de bits deseados.

Apagar el sistema

Cuando se presiona el pulsador de apagado el sistema operativo se apaga correctamente y se le quita la alimentación a la TS-7260.

En la Figura 5 se muestra un diagrama que explica como interactúan algunos programas del software del sistema embebido.

V. IMPLEMENTACIÓN DEL SOFTWARE DEL SISTEMA EMBEBIDO

Para tener la seguridad de que la SBC funcionara correctamente, se verificó el puerto serie, el puerto ethernet y algunos comandos básicos de GNU/Linux.

Luego se intentó bootear Debian Sarge 3.1 desde una tarjeta SD. Como el RedBoot que trae originalmente la placa TS-7260 no soportaba el booteo desde la tarjeta SD fue necesario cambiarlo por el TS-SDBOOT⁵[11].

El kit de desarrollo de Technologic System incluía el código fuente de Debian Sarge 3.1, para el kernel 2.4.26-ts10 que se cross-compiló para crear la imagen comprimida zImage. Para implementarlo exitosamente fue necesario modificar el código fuente proporcionado por Technologic System para obtener módulos de audio y tarjeta de almacenamiento, direccionamientos de memoria correctos y comando de ejecución de inicio del kernel; solucionar errores de dependencia y código mal implementado. Para poder solucionar estos inconvenientes fue necesario modificar archivos del fuente del kernel desrollados en C.

Para obtener los errores del sistema operativo se utilizó la herramienta strace que muestra las llamadas al sistema operativo. Analizando los distintos errores que se obtuvieron al utilizar dicha herramienta se depuro el código⁶.

Posteriormente se cross compilaron los módulos y bibliotecas necesarias para controlar las distintas interfaces de la SBC que traía incluido el kit de desarrollo. También fue necesario modificar los códigos fuentes de los cuales dependía el módulo de audio porque contenían diversos errores, como por ejemplo no se podía grabar correctamente.

Además de utilizar el TS-SDBOOT fue necesario particionar en 3 la tarjeta SD y realizar los siguientes procedimientos:

1ra partición.: Instalar el zImage del kernel de GNU/Linux deseado.

2da partición.: Instalar linuxrc y modificar algunos de los parámetros de dicho programa.

3ra partición.: Instalar Debian Sarge 3.1, nuevos módulos crosscompilados y archivos creados durante la realización del proyecto encargados de manejar la placa interfaz y grabar.

Para poder bootear Debian Sarge 3.1 desde una SD card fue necesario cambiar el RedBoot original de la TS-7260 por el TS-SDBOOT [11].

El kit de desarrollo de Technologic System incluía el código fuente de Debian Sarge 3.1, para el kernel 2.4.26-ts10 que se cross-compiló para crear la imagen comprimida zImage. Para implementarlo exitosamente fue necesario modificar el código fuente proporcionado por Technologic System para obtener módulos de audio y tarjeta SD, direccionamientos de memoria correctos y comando de ejecución de inicio del kernel; solucionar errores de dependencia y código mal implementado.

Posteriormente se cross compilaron los módulos y bibliotecas necesarias para controlar las distintas interfaces de la TS-7260 que traía incluido el kit de desarrollo. También fue

⁵Una vez realizada esta acción no es posible volver a bootear desde la SDRAM, a menos que se la envíe a la fábrica de origen; sin embargo, decidimos realizarla.

⁶Los archivos correctos se encuentran en: mosobo.it.com.uy/mosobo_2.4.26-ts10.tar.gz. Las herramientas para cross compilar el kernel se encuentran en: mosobo.it.com.uy/toolchain.tar.gz.

necesario modificar los códigos fuentes de los cuales dependía el módulo de audio ya que contenían diversos errores.

VI. TRATAMIENTO DE SEÑALES

Para realizar el reconocimiento de actividades se utilizó una representación apropiada de las señales acústicas y modelado estadístico que tiene su base en los Modelos Ocultos de Markov (HMMs)[12]. Cabe destacar que todo el análisis de las señales se realiza en un PC.

Antes de generar y entrenar los modelos las señales de audio debieron ser remuestreadas y filtradas por la presencia de ruido blanco en altas y bajas frecuencias. Para implementar el software de reconocimiento de actividades se utilizó el software Hidden Markov Model Toolkit (HTK)[13].

Con el fin de evaluar cualitativamente el resultado de grabar a los rumiantes con el dispositivo desarrollado y realizar el reconocimiento de eventos y actividades con el software implementado, definimos la probabilidad de acierto de la actividad A (simbolizando pastoreo, rumia o descanso) como:

$$Pa_A = \frac{100}{t_A} \sum_{i=1}^{K_A} \left(\sum_{j=1}^{N_i} t_{Aij} \right)$$

donde K_A es la cantidad de intervalos de la actividad A en la señal a evaluar, t_A es el tiempo total etiquetado de la actividad a evaluar A, N_i es la cantidad de intervalos de la actividad A en t_{Ai} detectados en el reconocimiento y t_{Aij} es el intervalo de tiempo j de la actividad A en t_{Ai} en el reconocimiento.

Los resultados obtenidos de los porcentajes de acierto a partir de una señal de 1 hora de duración fueron: $Pa_P=100\%$, $Pa_R=100\%$ y $Pa_D=52\%$. Tanto la rumia como el pastoreo se detectaron correctamente mientras que el descanso solo se detectó un poco más de la mitad del tiempo. Esto se debe a que cualquier ruido del ambiente puede ser considerado para el modelo, como un evento de rumia o arranque, por lo que parte del descanso puede confundirse con la rumia o el pastoreo.

VII. CONCLUSIONES Y PERSPECTIVAS

Utilizando la placa TS-7260 con Linux embebido se desarrolló un dispositivo capaz de grabar y almacenar correctamente los sonidos emitidos por bovinos y que puede permanecer a la intemperie junto con los mismos sin presentar problemas de funcionamiento. El dispositivo es de fácil colocación, no interfiere con el comportamiento de los animales, no les genera lesiones y además su costo, que ronda los 375 USD en materiales⁷, es relativamente bajo en comparación con otros dispositivos que se utilizan para este fin.

Se logró una independencia energética de 24 horas, resultando en un peso del dispositivo de 1700 g sin bozal. Utilizando las baterías de litio-ion mencionadas el peso total del dispositivo baja a 800 g sin bozal.

A pesar de haber tenido varios inconvenientes con las herramientas de desarrollo brindadas por el fabricante de la SBC (código fuente mal implementado, errores de direccionamiento

de memoria, dependencias, etc), se logró implementar exitosamente el software y sistema operativo del sistema embebido, cumpliendo con todos los objetivos planteados.

Utilizando la herramienta de desarrollo HTK se implementó un sistema de reconocimiento de eventos de las actividades que realizan los rumiantes; a partir del cual se implementó un sistema de reconocimiento de actividades (pastoreo, rumia y descanso).

Se lograron buenos resultados en el reconocimiento de las actividades de pastoreo y rumia (porcentajes de acierto del 100 %) mientras que el resultado de la actividad descanso no fue tan satisfactorio (porcentaje de acierto del 52 %).

En un futuro se le puede incorporar un GPS con el cual se podría relacionar la ubicación del animal al realizar cada actividad. Además, mediante el uso de mapas se podría determinar el tipo de pastura que ingirió el animal y estimar la materia seca consumida lo cual brinda información adicional del comportamiento ingestivo animal.

REFERENCIAS

- [1] P. Chilibraste, P. Soca, D. Mattiauda, O. Bentancur, and P. Robinson., "Short term fasting as a tool to design effective grazing strategies for lactating dairy cattle: a review." *Australian Journal of Experimental Agriculture*, 2007.
- [2] J. Hodgson, "The control of herbage intake in the grazing ruminant." *Hill Farming Research Organization*, 1985.
- [3] E. Sazonov, S. Schuckers, P. Lopez-Meyer, O. Makeyev, N. Sazonova, E. L. Melanson, and M. Neuman., "Non-invasive monitoring of chewing and swallowing for objective quantification of ingestive behavior." *Electronic Journals from Institute of Physics Publishing.*, 2008.
- [4] D. Milone, H. Rufiner, J. Galli, E. L. f, and C. Cangiano, "Computational method for segmentation and classification of ingestive sounds in sheep," *Science Direct*, 2009.
- [5] Gumstix, "Gumstix developer site - gumstix overo - feature overview," agosto 2009. [Online]. Available: <http://www.gumstix.net/Hardware/view/Hardware-Specifications/Overo-Specifications/112.html>
- [6] WinSystem, "Winsystem," agosto 2009. [Online]. Available: <http://www.pc104plus.com/products/PPM-LX800-G.cfm>
- [7] Winsystem, "Winsystem," agosto 2009. [Online]. Available: <http://sbc.winsystems.com/products/EPX-GX500.cfm>
- [8] T. Systems, <http://www.embeddedarm.com/products/board-detail.php?product=TS-7260>, agosto 2009.
- [9] BusyBox, "Busybox," junio 2010. [Online]. Available: <http://www.busybox.net/>
- [10] "Pelican products 1060 micro case," setiembre 2009. [Online]. Available: http://www.pelican.com/cases_detail.php?Case=1060
- [11] T. System, "Ts-boot," noviembre 2009. [Online]. Available: <http://www.embeddedarm.com/software/arm-linux-fastboot-ts7300.php>
- [12] "Hidden markov models," setiembre 2009. [Online]. Available: [http://jedlik.phy.bme.hu/\\$sim\\$gerjanos/HMM/node2.html](http://jedlik.phy.bme.hu/simgerjanos/HMM/node2.html)
- [13] U. de Cambridge, "htk3," noviembre 2009. [Online]. Available: <http://htk.eng.cam.ac.uk/>

⁷No incluye gastos de aduanas ni envíos.