

Experiencias de Análisis de Consumo Energético en Redes de Sensores

Medina Santiago¹, Romero Fernando¹, De Giusti Armando^{1,2}, Tinetti Fernando G.^{1,3}

¹Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática
Universidad Nacional de La Plata, 50 y 120 2do piso, La Plata, Argentina

²CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas

³CIC – Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires
{smedina, fromero, degiusti, fernando}@lidi.info.unlp.edu.ar

Abstract. En este artículo se presenta la experimentación para evaluación de consumo energético de microcontroladores y módulos de comunicación utilizados o propuestos en el mercado para la construcción de y operación con redes de sensores inalámbricas o Wireless Sensor Network (WSN). Se identifican y analizan tanto los denominados modos de períodos de consumo mínimo (o *sleep mode/s*) de los módulos/hardware utilizado (en los no se transmiten datos) como los períodos en los cuales se transmiten datos se necesita y consume energía. Estos consumos se deben identificar tanto en los microcontroladores como en los propios módulos de comunicación específicos utilizados, dado que una gran cantidad de WSN combina ambos tipos de dispositivos en (inter)operación. En particular, los módulos de comunicación inalámbrica son utilizados para la transmisión propiamente dicha de los datos, utilizando diferentes estándares y protocolos, como WiFi, Bluetooth, ZigBee, LoRa, etc. Este tipo de redes generalmente tienen un despliegue en ambientes en donde no existe alimentación eléctrica constante por lo tanto es necesario optimizar el consumo energético de cada nodo ya que deben ser alimentados con baterías con una capacidad limitada.

Keywords: Red de Sensores, LoRa, WiFi, WSN.

1 Introducción

Las redes de sensores inalámbricos constituyen uno de los sistemas de procesamiento de información con mayor crecimiento y ámbito de aplicación de los últimos tiempos. Son denominados usualmente también como WSN (Wireless Sensor Network) o WSAN (Wireless Sensor & Actuator Network) [1] [2], y son un concepto alternativo de las MANET (Mobile Ad hoc NETWORK), enfocadas directamente en la interacción con el ambiente en donde son desplegadas, en lugar de la interacción con las personas.

El inicio de la evolución de las redes de sensores inalámbricos está dado principalmente por diferentes aplicaciones militares. Entre algunos de los primeros proyectos con similitudes a las actuales características de las WSN se pueden nombrar: el proyecto Chain Home, un anillo de radares para alerta temprana que podía detectar y rastrear

aeronaves durante la Segunda Guerra Mundial; el proyecto SOSUS, del acrónimo Sound Sourviellance System, utilizado en la guerra fría por el ejército de Estados Unidos para rastrear submarinos soviéticos y el proyecto NORAD (North American Aerospace Defense Command), también desarrollado durante la guerra fría, para en control y la defensa aérea en los Estados Unidos.

En la estructura básica de una Red de Sensores se pueden definir 3 partes: los nodos sensores, la red inalámbrica de comunicación y nodos centralizadores o gateways [3]. Los nodos sensores y los centralizadores están basados en pequeños dispositivos de hardware, los microcontroladores, que utilizan transceptores inalámbricos para comunicación y un módulo de alimentación.

El crecimiento exponencial en el desarrollo de hardware del último tiempo ha generado nuevos microcontroladores de pequeño tamaño, con mayor capacidad de procesamiento y menor consumo energético. Estos dispositivos, combinados con el avance sobre nuevas tecnologías de comunicación inalámbrica dan como resultado la capacidad de desplegar diferentes tipos de nodos que conforman Redes de Sensores aplicadas en diferentes ámbitos, para obtener datos y realizar operaciones. Entre las características de las redes de sensores se pueden identificar: Baja tasa de transferencia, Limitación en la alimentación o bajo consumo de los nodos sensores, alimentación por baterías, escalabilidad, gran cantidad de nodos. Estas Redes tienen diferentes ámbitos de aplicación debido a que dan la posibilidad de desarrollar los nodos de manera personalizada para su utilización en cada área específica, y adaptar la topología de red a las necesidades del problema. Entre los ejemplos más comunes de aplicación de las Redes de Sensores Inalámbricos están, desarrollos militares, monitoreo ambiental, agricultura, seguridad, domótica y ciudades inteligentes [3] [4] [5].

Uno de los aspectos importantes a considerar en el desarrollo de los nodos de una red de sensores es el consumo energético. Dadas las características de los ambientes en donde son desplegadas este tipo de redes, no se cuenta en la mayoría de los casos, con una conexión a la red eléctrica que sea constante. Por lo tanto, es necesaria la utilización de baterías que se encarguen de proveer alimentación a cada nodo de la red. Si bien el avance en el desarrollo informático/electrónico dio como resultado microcontroladores y tecnologías de comunicación de bajo consumo, es necesario hacer un análisis preciso en donde se combinen diferentes aspectos a considerar:

- La elección de los componentes electrónicos que conforman a un nodo.
- El tipo de transmisor/receptor de radio.
- Las posibilidades de modos de ahorro de energía que da el microcontrolador.
- El tipo de batería.

Este trabajo es una continuación de la experimentación realizada en [6], haciendo foco en el consumo energético de los nodos que componen una red de sensores inalámbricos y las técnicas disponibles para reducir dicho consumo.

2 Consumo Energético en una WSN

Los principales campos de aplicación de las WSN requieren que los nodos que la conforman tenga autonomía energética, y esto se logra utilizando pilas o baterías (algunas veces recargadas con paneles solares). Estos componentes electrónicos de alimentación

tienen su duración en función del consumo que se les requiere, y por lo tanto es de vital importancia realizar un análisis de las características de consumo energético de cada nodo de la red para poder estimar la autonomía de funcionamiento sin necesidad de un cambio o recarga en las baterías.

Los componentes principales de un nodo son el microcontrolador y el transceptor inalámbrico para transmisión. Para lograr una estimación correcta de consumo se suele realizar un análisis individual de cada parte y su funcionamiento en conjunto. En el caso del microcontrolador existen dos aspectos de análisis. El primero es a nivel de hardware, en el que por ejemplo se evalúan los componentes electrónicos necesarios para nivelar la alimentación de entrada (un regulador de tensión) en función de los niveles que maneja la batería utilizada. También en este nivel hay que considerar las optimizaciones de hardware la temperatura del ambiente, ya que este factor podría hacer que los componentes electrónicos se vean afectados y requieran un mayor consumo para funcionar correctamente. El segundo aspecto a evaluar son las opciones de optimización energética que se pueden lograr a nivel de software, en este aspecto entran los modos *sleep* o modos bajo consumo, la posibilidad de variar la velocidad (frecuencia) del reloj del procesador en ejecución y la desactivación de componentes que no se utilicen.

Por el lado del transceptor de radiofrecuencia, el consumo va a depender de la tecnología que se utilice, existen varias de baja consumo como ser LoRa o ZigBee. También hay disponibles tecnologías que no son de bajo consumo como WiFi o Bluetooth. En el análisis de consumo a nivel comunicación es importante determinar la cantidad de paquetes que necesitan ser enviados por el nodo en un tiempo específico, ya que a mayor cantidad de paquetes habrá mayor consumo. Otro punto a considerar es la potencia que se utiliza en la transmisión, donde se debe lograr una optimización entre la distancia y la potencia, de tal manera que se configure la transmisión con la menor potencia posible (que implica menor consumo), pero que alcance la distancia necesaria para que el nodo siga siendo parte activa de la red.

3 Hardware Utilizado

A los fines prácticos del análisis de este artículo se eligieron placas de desarrollo y módulos que son de alta disponibilidad y que ya mostraron buen funcionamiento como parte de una WSN en la experimentación realizada en [6].

3.1 NodeMCU ESP8266

NodeMCU es un firmware y una placa de desarrollo de código abierto, basada en el chip de la empresa Espressif ESP8266-12E. La placa agrega al microcontrolador el chip para la conexión UART a través de un puerto micro USB, que sirve como alimentación y punto de comunicación [7]. El ESP8266 es un microchip WiFi de bajo consumo que contiene como CPU al procesador Tensilica LX106 de 32-bit RISC, a 80MHz; con 64 KB de memoria de instrucciones y 96KB de datos [8] [9]. Tiene integradas varias características de hardware que lo hacen apropiado para una variedad muy amplia de aplicaciones: 1 ADC de 10-bit, 17 GPIO pins, Interfaces: 3 SPI, 1 I2C, 1 I2S y 2 interfaces

UART, PWN, Protocolo WiFi 802.11 b/g/n, entre otras. Las dos características más importantes, sin embargo, no son necesariamente de hardware: bajo costo y amplio soporte de programación. En particular, la programación utilizando el IDE y las bibliotecas de Arduino han generado una gran cantidad de ejemplos disponibles gracias al uso masivo. En la Fig. 1 se puede apreciar, por ejemplo, que los pines GPIO son identificados como los del Arduino UNO (D0, D1, etc.).



Fig. 1 – Placa de desarrollo NodeMCU

El ESP8266 tiene predefinidos 3 tipos de modos sleep y un consumo teórico para cada uno de ellos [10], tal como se muestra en la Fig. 2. Cada uno afecta a diferentes partes de la electrónica del micro.

Item	Modem-sleep	Light-sleep	Deep-sleep
Wi-Fi	OFF	OFF	OFF
System clock	ON	OFF	OFF
RTC	ON	ON	ON
CPU	ON	Pending	OFF
Substrate current	15 mA	0.4 mA	~ 20 μ A
Average current	DTIM = 1	16.2 mA	1.8 mA
	DTIM = 3	15.4 mA	0.9 mA
	DTIM = 10	15.2 mA	0.55 mA

Fig 2 - Tabla de Modos Sleep ESP8266

3.2 Heltec WiFi LoRa 32

La Fig. 3 muestra la placa de desarrollo construida por la empresa Heltec. Esta placa incluye dos chips principales, el ESP32 y el transceptor LoRa SX1278 (con conector I-PEX, para antena externa), sumados a un convertidor CP210x MicroUSB a UART, una pantalla 0,96" OLED y un conector Molex PicoBlade de 2 pines para batería LiPo/Li-Ion [11]. El módulo WiFi puede transmitir a una tasa de hasta 150Mbps. El módulo Bluetooth, tiene funcionamiento normal y también BLE (Bluetooth Low Energy).

El ESP32 es una solución integrada de WiFi-Bluetooth con radiofrecuencia de 2.4GHz. Basado en el microprocesador Tensilica Xtensa LX6 de 32 bit, a 160MHz, con 448 KB de memoria ROM y 520 SRAM [12] [13]. Algunas de sus características más

importantes de hardware son: 34 GPIOs programables, 1 SAR ADC de 12bit, 2 DAC de 8 bit, varias interfaces (SPI, I2S, I2C), PWM y Protocolo 802.11 b/g/n. EL SX1278 es un chip transceptor de radio frecuencia LoRa desarrollado por Semtech, dentro de la placa de desarrollo de Heltec, tiene dos versiones de frecuencia: 433MHz y otro que soporta las frecuencias 868MHz y 915MHz [14]. Este módulo se conecta vía SPI al ESP32 (conexión integrada en la propia placa).



Fig. 3 – Placa de Desarrollo de Heltec

El ESP32 tiene 4 modos sleep que desactivan los componentes internos del micro según explica la tabla de la Fig. 4.

Power mode	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
Sleep pattern	Association sleep pattern			ULP sensor-monitored pattern	-
CPU	ON	ON	PAUSE	OFF	OFF
Wi-Fi/BT base-band and radio	ON	OFF	OFF	OFF	OFF
RTC memory and RTC peripherals	ON	ON	ON	ON	OFF
ULP co-processor	ON	ON	ON	ON/OFF	OFF

Fig 4 -Tabla de Modos Sleep ESP32

3.3 Adafruit INA219

Para la medición del consumo energético se seleccionó el sensor de corriente Adafruit INA219 como el que se muestra en la Fig. 5. Este sensor es de característica invasivo ya que se conecta en serie al elemento y de esta manera se obtiene la corriente que pasa por ese conductor. El módulo sensor está compuesto por un amplificador de precisión que mide el voltaje que cae en la resistencia de detección de 0.1ohm que tiene la placa. Dado que la diferencia máxima de entrada del amplificador es $\pm 320\text{mV}$, da un rango de medición de hasta $\pm 3.2\text{A}$. Con el ADC interno de 12bits, la resolución en el rango $\pm 3.2\text{A}$ es $0,8\text{mA}$. Con la ganancia interna establecida al mínimo, la corriente varía entre $\pm 400\text{mA}$ y la resolución es de $0,1\text{mA}$. Se puede alimentar con 5V o 3V y maneja una comunicación I2C para la salida de los datos medidos.

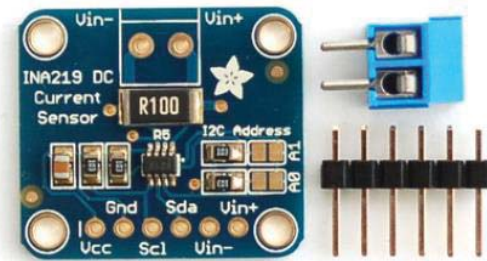


Fig 5 - Placa del Sensor INA219 [15]

4 Experimentación

En todas las pruebas de la experimentación se utilizó el mismo sistema para la medición de consumo [16], mostrado en la Fig. 6: se captura la corriente con el sensor INA219 y se envía por I2C a un Arduino UNO [17]. En el código que corre el Arduino se utilizó la librería que fue desarrollada por el fabricante del módulo sensor (Adafruit) para esta plataforma. La alimentación al microcontrolador del cual se censa el consumo fue a través del puerto micro USB de cada placa con un voltaje constante de 5V.

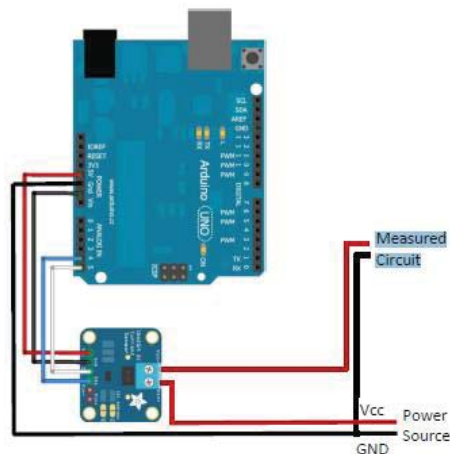


Fig. 6 - Esquema de conexión de la experimentación [15]

La experimentación se dividió en dos partes en función de la tecnología de comunicación estudiada en cada caso. Primero se hicieron experimentos de la conexión WiFi en cada una de las plataformas de desarrollo, en donde también se consideran los modos sleep, y segundo en la conexión inalámbrica LoRa.

I-WiFi

Los escenarios planteados fueron similares para cada placa de desarrollo, NodeMCU ESP8266 y Heltec WiFi LoRa 32. Todos se desarrollaron en scripts dentro del IDE

Arduino de programación:

- Se analizó la transmisión y recepción continua de paquetes WiFi.
- Se analizaron los modos sleep específicos de cada una de las placas y sus posteriores procesos de despertar.
- Se analizó el proceso de conexión de un cliente a una red WiFi.

Como agregado de las mediciones recién listadas, se consideraron las 4 variaciones que presenta el ESP8266 al momento de despertarse del modo DeepSleep:

- WAKE_RF_DEFAULT: cuando se reinicia el microprocesador, únicamente se calibra la señal de radio si el chequeo da error.
- WAKE_RFCAL: cuando se reinicia el microprocesador siempre se calibra la señal de radio.
- WAKE_NO_RFCAL: cuando se reinicia el microprocesador no se calibra la señal de radio.
- WAKE_RF_DISABLED: cuando se reinicia el microprocesador se deshabilita la señal de radio (como en el modo modem sleep). No permite ni enviar ni recibir datos vía WiFi.

II – LoRa

En el caso de LoRa, los experimentos se realizaron solamente sobre la placa de Heltec, ya que cuenta con el transceptor LoRa SX1278. Para el desarrollo de los códigos de prueba se utilizó el IDE Arduino, con la última versión de la librería desarrollada por Heltec para las comunicaciones LoRa de su placa. En este caso las pruebas se dividieron entre las mediciones de recepción y las de transmisión de paquetes. En transmisión, la nueva versión de la librería permite variar el factor de potencia de la comunicación inalámbrica (entre 0 y 20), para lograr un mayor alcance. Por lo tanto se probaron cada uno de los factores disponibles.

5 Resultados

Se muestran en este punto los resultados de los experimentos comentados en la sección anterior con las muestras obtenidas en cada caso. En la conexión WiFi para cada tipo de placa utilizada se muestran los modos de operación como Access Point y como cliente respetivamente. Finalmente se muestran los resultados correspondientes al módulo de comunicaciones LoRa en recepción y transmisión. En la siguiente sección se mencionan las consideraciones como análisis de los resultados obtenidos.

5.1 NodeMCU

Las Tablas 1 y 2 a continuación muestran los resultados obtenidos para la placa de desarrollo NodeMCU. Los modos de sleep corresponden a cada modo de operación. Los picos de consumo se producen en períodos relativamente cortos de tiempo en algunos estados de operación del módulo.

Tabla 1. Consumo de cada por estado en modo Access Point

Operación	Consumo promedio (mA)	Consumo pico (mA)
Transmisión		360
Recepción	76	
ModemSleep	15	
DeepSleep	0.55	
DeepSleep - WAKE_RF_DISABLED	14	
DeepSleep - WAKE_RFCAL	76	360
DeepSleep - WAKE_NO_RFCAL	70	310
DeepSleep - WAKE_RF_DEFAULT	72	320
LightSleep	2	

Tabla 2. Consumo por estado en modo cliente

Operación	Consumo promedio (mA)	Consumo pico (mA)
Conexión		330
Conectado	68	
Transmisión		270
DeepSleep	0.6	
ModemSleep	15	

5.2 Heltec WiFi LoRa 32

Las Tablas 3, 4, y 5 a continuación muestran los resultados obtenidos para la placa de desarrollo de Heltec. Los modos de sleep corresponden a cada modo de operación WiFi. Los picos de consumo se producen en períodos de tiempo cortos en algunos estados específicos.

Tabla 3. Consumo por estado en modo Access Point

Operación	Consumo promedio (mA)	Consumo pico (mA)
Transmisión		310
Recepción	150	
Inicio	130	280
DeepSleep	10	
LigthSleep	11	
Hibernation	10	
ModemSleep	15	

Los valores de la operación de Inicio que muestra la tabla anterior son de los milisegundos que tarda el modulo en iniciar y activar la red WiFi como Access Point, durante ese proceso llega a picos de 280mA que corresponden a la calibración de la antena WiFi.

Tabla 4. Consumo por estado en modo cliente

Operación	Consumo promedio (mA)	Consumo pico (mA)
Conexión		150
Conectado	56	
Transmisión		162
DeepSleep	10	
ModemSleep	15	

Tabla 5. Consumo por estado utilizando el transceptor de LoRa

Operación	Consumo promedio (mA)	Consumo pico (mA)
Recepción	80	
Transmisión Tx Power 0		102
Transmisión Tx Power 5		107
Transmisión Tx Power 10		114
Transmisión Tx Power 15		132
Transmisión Tx Power 20		177

6 Conclusiones y Trabajo Futuro

Este trabajo se basa en una experimentación para la obtención de valores de referencia del consumo energético de dos placas de desarrollo con dos tecnologías de comunicación inalámbrica distintas, para tomar decisiones en cuanto al armado del nodo de una red de sensores. Los resultados obtenidos muestran que ambas placas de desarrollo presentan herramientas efectivas para lograr un consumo global bajo. Considerando que en una aplicación de red de sensores el tiempo concreto de funcionamiento es muy poco y la placa pasaría la mayoría del tiempo en un estado de reposo (modo sleep) con su transceptor inalámbrico desactivado, por lo tanto, son una buena elección a la hora de desarrollar un nodo basándose en cada una de ellas.

En la comparación con los valores teóricos que dan los fabricantes se puede ver que los resultados obtenidos están por encima. Esto se debe a que los valores teóricos están definidos para el chip solo, y en nuestro caso de experimentación estamos trabajando sobre una placa de desarrollo entera, que incluye electrónica extra para la regulación de tensión, el manejo de la conexión USB y leds de encendido, entre otros componentes.

Respecto a la placa NodeMCU se obtiene un excelente valor de consumo al llevarla al modo de DeepSleep, en ambos modos de operación WiFi; esto es muy útil si se tienen conexiones con transmisiones esporádicas. Se debe considerar como muestran las tablas que tanto las transmisiones, como la conexión a la red inalámbrica tienen picos altos de consumo, que duran milésimas de segundo, por lo tanto la batería que alimenta a esta placa debe poder resistir estos picos.

En la placa de Heltec puede llegar a dos conclusiones. La primera respecto a las conexiones WiFi, que muestran una leve ventaja frente a las de la placa anterior respecto a la transmisión y recepción de paquetes, pero no presenta mejores valores en los modos

sleep. Y la segunda conclusión se da con los valores obtenidos de las pruebas realizadas da la comunicación LoRa, cabe aclarar que estos valores solo cuentan el consumo en estado básico de la placa (con la conexión WiFi desactivada) para hacer foco en el transceptor SX1278. Se puede ver en los resultados que hay una curva creciente en consecuencia directa al aumento de la potencia de transmisión. Solo se consideraron los valores de pico ya que el valor promedio siempre se mantuvo alrededor de los 80mA como en el estado de recepción, pero se deben considerar estos picos para elegir el módulo de alimentación correcto.

Como trabajos futuros se plantean las pruebas específicas de las distancias de funcionamiento de LoRa en las variaciones de la potencia de transmisión y la posibilidad de programar a más bajo nivel el módulo ESP8266 para variar la potencia en las transmisiones WiFi. También es de interés para complementar el análisis funcional de una WSN realizar pruebas de performance a nivel de seguridad y encriptación de datos con las placas utilizadas en este artículo.

Bibliografía

1. Estándar IEEE 802.11 Wireless Local Area Network, <http://www.ieee802.org/11/>
2. Estándar IEEE 802.15 Wireless Personal Area Network, <http://www.ieee802.org/15/>
3. Sohraby, K., Minoli, D., Znati T.: Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons (2007).
4. Raghavendra, C. S., Sivalingam, K. M., Znati, T.: Wireless sensor networks. Springer (2006).
5. Akyildiz, I., Vuran, M.: Wireless sensor networks(Vol. 4). John Wiley & Sons (2010).
6. Medina, S., Romero, F., De Giusti, A. E., & Tinetti, F. G. (2018). Análisis para despliegue de una red de sensores heterogénea. In XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018).
7. Documentación NodeMCU, www.nodemcu.com
8. ESP8266EX Datasheet Version 5.8, Espressif Systems. Año 2018. Disponible en https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
9. ESP8266 Technical Reference Version 1.3, Espressif Systems. Año 2017. Disponible en https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf
10. ESP8266 Low Power Solutions Version 1.1, Espressif Systems. Año 2016. Disponible en https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf
11. Documentación WIFI LoRa 32, <http://www.heltec.cn>
12. ESP32 Datasheet Version 2.3, Espressif Systems. Año 2018. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

13. ESP32 Technical Reference Manual Version 3.4, Espressif Systems. Año 2018.
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
14. SX1276/77/78/79 Datasheet Rev 5, Semtech. Año 2016.
https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf
15. Adafruit INA219 Current Sensor Breakout, Adafruit Systems. Año 2019.
Disponible en <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ina219-current-sensor-breakout.pdf>
16. Fernando G. Tinetti, Santiago Medina, Mariano Méndez, Fernando L. Romero, “Energy Measurements in IoT/ Maker Development Cards”, III-LIDI, Facultad de Informática, UNLP, Sept. 2016, Technical Report TR-RT-01-2016, disponible en <http://fernando.bl.ee/>
17. Arduino LLC, “Arduino – ArduinoUnoRev3”,
<https://store.arduino.cc/usa/arduino-uno-rev3>