

Raspberry Pi: uso potencial en astronomía

Rubén Díez Lázaro

Clube Vega

I Workshop CMON
15 de Diciembre de 2012

Computación, Electrónica y Astronomía

Computación y Astronomía

- Tratamiento de imágenes.
- Obtención de imágenes.
- Cálculos y efemérides.
- Control de telescopios, cúpulas, ...

Electrónica y Astronomía

- Automatización.
- Sensores.
- Control de telescopios, cúpulas, ...

Punto de vista del aficionado

El aficionado que aborda la Computación y la Electrónica como herramienta en Astronomía puede ocupar uno de estos nichos:

Nichos que puede ocupar el aficionado:

- **Usuario final:** Usa software de retoque fotográfico, preparación y registro de observaciones (computación). Gadget electrónicos comprados (Electrónica).
- **Cacharreador:** Escribe software (Computación). Crea Gadgets de "alto nivel" (Electrónica).
- **Básico:** Crea componentes básicos... perfil casi inexistente a nivel aficionado.

Estado del arte

Computación

Software para PC y para Smartphones: Procesado de imágenes. Observatorios remotos.

Electrónica

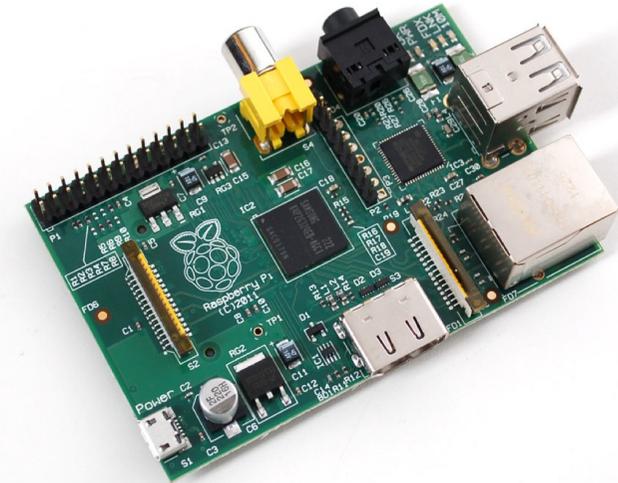
Microcontroladores: Control de telescopios. Observatorios remotos.

¿Y SI PUDIÉRAMOS UNIR LAS DOS COSAS??: Sistemas embebidos.

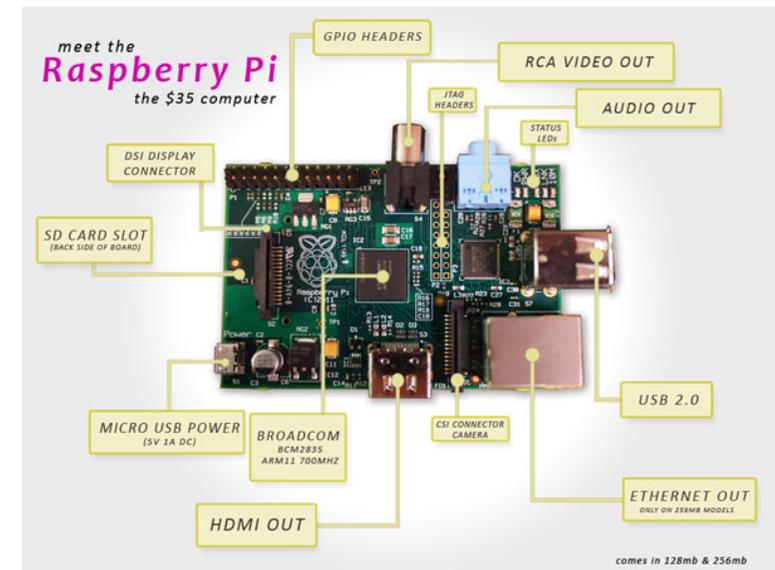
Si unimos la parte *electrónica* con la parte computacional tenemos un nuevo concepto: Los **sistemas embebidos**.

Características

- Se trata de sistemas que se aplican a “*uso final*”.
- Interaccionan con el usuario como un computador de propósito general.
- Hasta ahora el problema para hacer sistemas embebidos DIY era la parte computacional. El gran avance en tecnologías relacionadas con los Smartphones está acabando con este problema.
- El Raspberry Pi es un sistema computacional con unas características excelentes para su uso en sistemas embebidos DIY.



- Desarrollada por una fundación caritativa.
- Pensado para la enseñanza de las ciencias computacionales.
- Dos modelos: A y B; el B es más potente. . .
- Basado en procesadores ARM, del tipo usado en los móviles.
- Conectores para entrada (CSI) y salida (DSI) directa de vídeo.
- Conector de propósito general: GPIO.
- Sistemas operativos *oficiales*: Debian, Arch, Risc OS.
- Las imágenes de SO se bajan y vuelcan a un SD con un PC.



- No tiene RTC ni interruptor on/off.
- Si el dispositivo USB absorbe potencia, se necesita hub USB alimentado.
- Aceleración de vídeo por hardware.

- Usar hub USB alimentado para HD externo y captador analógico de vídeo.
- Sólo almacena datos, no los procesa.
- Conexión de red para NTP y recoger los datos.
- Software adaptado a la Raspberry Pi.

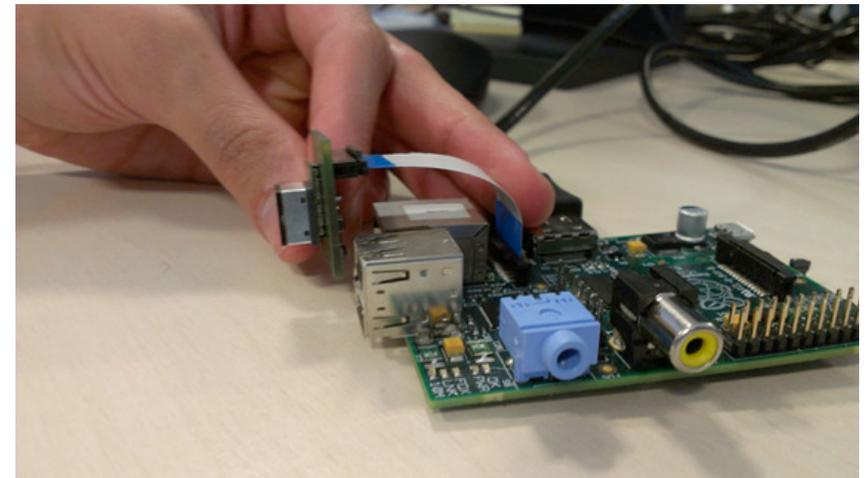
- Cámara USB, CSI o sensor específico conectado por USB.
- Conexión de red para NTP y recoger los datos.
- Software adaptado a la Raspberry Pi.

- Circuito auxiliar para conectar a la montura. Controlado a través de GPIO.
- Cálculo de movimiento por software: Correcciones y seguimiento de objetos no estelares.
- Interfaz por red. Cliente en Android.
- Software adaptado a la Raspberry Pi.

- Circuito auxiliar para control de paso a paso, conectado a la GPIO.
- Cálculo de movimiento por software: adaptable a ecuatoriales, altazimutales, seguimiento de objetos no estelares.
- Interfaz por red. Cliente en Android.
- Diseño modular de componentes.

- Circuitos auxilios, conectado a la GPIO. USB. Red.
- Control de cúpula, sensor de lluvia, . . .
- Interfaz por red. Cliente en Android.
- Diseño modular de componentes.

- Posibilidad de cámara USB o directamente por CSI.
- Adaptar software de seguimiento.
- Interacción con el sistema de control de telescopio.
- Interfaz por red. Cliente en Android.



- Existen otras posibilidades:
http://elinux.org/RaspberryPi_Comparison
- Familiarizarse con el sistema.
- Explorar uso de GPIO.
- Compilar componentes software que ya existen (librerías).
- Adaptar y/o desarrollar software.
- Aprender a realizar placas de circuito (PCB) con material casero.

- Grandes promesas...
- Mucho trabajo.
- Apuesta por los sistemas embebidos en Astronomía.
- Liberación de la esclavitud del PC
- Colaboración con otros aficionados.

Cosas que tal vez no sepamos:

- Se trata de una arquitectura ARM: no es compatible a nivel binario con x86.
- Diferencia entre programa *compilado* y programa *interpretado*...
- Construir proyectos grandes a partir de componentes sencillos: desarrollar de 0 sólo las partes necesarias.
- Separar la interface de usuario (UI) de la parte funcional.

Errores inducidos por la esclavitud del WinPC

- Diferencia entre el código fuente que, si está bien escrito, se puede portar a otras arquitecturas y la versión binaria de los programas, que no se pueden portar.
- La esclavitud del WinPC hace que se esté usando tecnología obsoleta.
- Los “*programas interactivos*” no son la mejor opción... salvo que no exista otra.
- Tal vez está empezando una nueva era de la computación en la que se están dejando atrás los viejos errores: la Raspberry Pi sería algo así como el nuevo *Spectrum*.

- **Página de la Raspberry Pi:**
<http://www.raspberrypi.org>
- **Linux embebido para la Raspberry Pi:**
http://elinux.org/RPi_Hub
- **The MagPi Magazine** (muy recomendado):
<http://www.themagpi.com>
- Un montón de vídeos demostrativos en YouTube.

Gracias por su atención
¿Preguntas?