CURSO: ROBÓTICA. CONECTANDO CON EL MUNDO FÍSICO

1 Nombre, apellidos y correo electrónico de EducaMadrid del participante

Antonio Sánchez Ruiz
Antonio.sanchezruiz@educa.madrid.org

2 Título de la Unidad Didáctica

Instalación y uso de s2a (Scratch 2.0 conectado a Arduino). Proyecto de ejemplo: mando de juego

3 Curso

3º E.S.O.

4 Descripción del proyecto

Para dar los primeros pasos en computación física con Arduino, y dado que en 1º y 2º van a aprender a programar con Scratch, se plantea el uso de Scratch for Arduino (**s4a**). El inconveniente principal que tiene es que está basado en la versión 1.4 de Scratch. Actualmente está disponible la 2.0, tanto en su versión off-line como, sobre todo, la versión on-line, y es la versión más difundida. Cualquier programa o juego de la página está en la versión 2.0. Me parece, pues, que usar el s4a tiene este inconveniente. Investigando, he hallado que existen dos alternativas para poder trabajar con la versión 2.0 y Arduino:

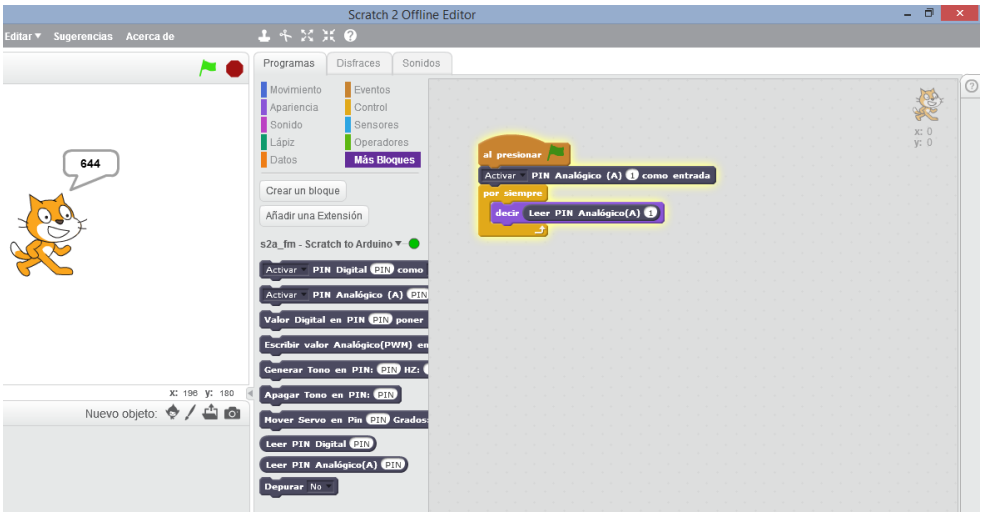
- hay una página que convierte proyectos de la versión 2.0 a 1.4, siempre y cuando no tenga órdenes exclusivas de la versión nueva. En todo caso, da error y señala cuál es la orden que no puede convertir. Se supone que quitándola funcionaría, aunque esto puede "descoyuntar" el programa. El enlace es: http://kurt.herokuapp.com/20to14/converted.sb
- la extensión **s2a_fm**, que funciona sobre Scratch 2.0, aunque su instalación y puesta en funcionamiento no es nada sencilla. Para instalarlo, he seguido las instrucciones de :
https://github.com/MrYsLab/s2a_fm/blob/master/documentation/Espa%C3%B1ol/s2a_fm_Espanish_tutorial.pdfy los archivos necesarios se pueden descargar de:
https://github.com/MrYsLab/s2a_fm

Al final de este documento añado un anexo con la explico de forma detallada la instalación, los problemas que me han surgido y cómo los he resuelto. Pero en resumen se trata de:

- Instalar Python
- Instalar Pyserial, que es un programa en python que permite la comunicación por el puerto serie.
- Instalar Pymata. Otro programa en Python, pero esta vez sin ejecutable.

Una vez instalado todo, hay que iniciar el programa s2a_fm desde la línea de comandos cada vez que queremos usar Scratch con la placa Arduino. También lo explico con detalle en el anexo.

Y por último, abrir el archivo s2a_fm_base_Es.sb2 en Scratch, en el que aparecen los bloques para controlar Arduino



Construcción de un mando a distancia de juego para Scratch.
Esta es la parte física del proyecto. Se trata de hacer un mando de juego sencillo con un potenciómetro, que servirá para controlar un sprite de izquierda a derecha, arriba-abajo, o bien su giro (sería un joystick, pero unidireccional, para juegos tipo ping-pong, Space Invaders,...) Además, se añade un pulsador que servirá para el "disparo". La conexión del potenciómetro se hace a través de la entrada analógica A1, y la del disparador con un montaje en pull-down con el pin digital 13 (para así comprobar también su funcionamiento con el LED de la placa). Se montan tal y como aparece en el curso. Los esquemas y fotos figuran abajo.

Existe un problema añadido a los expuestos. La idea inicial era poder aprovechar el mando para cualquiera de los juegos que existen ya hechos en la página de Scratch. El problema es que s2a_fm añade unos bloques para poder controlar Arduino, pero no se instalan con "carácter retroactivo". Es decir, aparecen cuando abro un archivo .sb2 que los trae ya (en concreto el s2a_fm-master\ScratchFiles\ScratchProjects\s2a_fm_base_Es.sb2). Si trato de abrir uno que no esté hecho a partir de ese inicial, esos bloques de control no aparecen.
La solución para poder aprovechar los ejemplos de la página, o proyectos ya hechos por los alumnos, es "Guardar a un archivo local" cada uno de los sprites, fondos,...(la opción está en el menú contextual de cada uno de estos objetos). Una vez guardados todos, hay que importarlos en un archivo abierto desde el original que posee los comandos de Arduino. Y a esta importación hay que añadir el hecho de que hay que crear a mano las variables que tuviera el juego original, ya que estas no se pueden guardar en el disco. En fin, que no es una solución buena para juegos que estén hechos con anterioridad.
Se plantean una serie de mejoras y retos:
- Con el mismo esquema, usar un cable de telefonía de 0'5 m o más, y una caja, tupper o tarrina en la que acoplar el pulsador y el potenciómetro a modo de mando a distancia, en lugar de en la protoboard.
- añadirle un motor con una excéntrica para que funcione como vibrador. Debería funcionar cuando se pierda la partida, o cuando te alcance una bala,...
- añadirle un par de leds, uno verde cada vez que dispares, uno reojo cada vez que aciertes,...
- Hacer un joystick de dos ejes, con dos potenciómetros, para poder jugar al resto de juegos.

5 Competencias que se trabajarán

Aprender a aprender. Desarrollado durante todo el método de trabajo.
Competencia matemática y competencias básicas en ciencia y tecnología, y la competencia digital.
Aplicar el conocimiento sobre el hardware, el software y los conceptos electrónicos para realizar tareas con una placa Arduino.
Aplicar los conocimientos sobre el funcionamiento de los sistemas informáticos para seleccionar el ordenador y el software más adecuados a las necesidades reales puntuales, aplicándolo a las necesidades reales creadas por una dificultad.
Competencias sociales y cívicas.
Relacionar los avances de la tecnología electrónica con el desarrollo de los ordenadores y sus repercusiones económicas y sociales.
Sentido de iniciativa y espíritu emprendedor y la conciencia y expresiones culturales.
Relacionar la historia de la computación y de las máquinas con la necesidad de facilitar las tareas rutinarias y con la disminución del tiempo necesario para llevarlas a cabo.

6 Objetivos

Conocer cómo es una placa de Arduino, sus tipos y el conexionado.
Introducir la programación del código fuente para el funcionamiento, la compilación y la ejecución.
Identificar los elementos de los circuitos y describir sus cometidos.
Llevar a cabo montajes a partir de una idea.
Analizar las características de los circuitos a través de su esquema simbólico.
Tomar conciencia de los riesgos derivados de la electricidad y respetar las normas de seguridad eléctrica.
Conocer las posibilidades que ofrece el trabajo con robótica.
Aprender el manejo de las herramientas más usuales en robótica.

7 Contenidos

Identificación del ordenador y la robótica como herramienta que facilita las tareas de la vida cotidiana y las múltiples posibilidades, así como su aplicación para el paso de lo teórico y digital a lo mecánico.
Identificación de las partes del circuito y de la placa.
Uso de los elementos en un circuito electrónico.
Manejo de los operadores en un circuito eléctrico y su conexionado.
Creación o modificación del código fuente.
Necesidad de trabajar con tutoriales y la consulta como elementos de aprendizaje.
Identificación de elementos a través de sus símbolos.
Interpretación de esquemas eléctricos elementales.
Montaje de circuitos sencillos a partir de un esquema.
Identificación de los riesgos de la energía eléctrica.
Importancia de la robótica en la actualidad.
Valoración de la utilidad de la robótica como herramienta de trabajo e interface entre lógica y mecánica.
Interés por el manejo de la robótica y la programación.
Cuidado del material de trabajo.
Respeto por el trabajo de otros compañeros.
Sensibilidad frente al ahorro de energía eléctrica.
Interés por comprender el funcionamiento de los elementos eléctricos y de los circuitos.
Observancia de las normas y medidas de seguridad al utilizar los aparatos eléctricos.

8 Criterios de evaluación

	Utiliza y gestiona un ordenador bajo un sistema operativo Windows Instala y desinstala de manera segura software Utiliza adecuadamente los dispositivos electrónicos como fuente de información y para crear contenidos Analiza los diferentes niveles de lenguaje de programación Utiliza con destreza un entorno de programación por bloques Es capaz de usar la línea de comandos de windows, y cambiar propiedades avanzadas de archivos. Describe las fases y procesos del diseño de proyectos tecnológicos Realiza búsquedas de información relevante en Internet Determina y calcula los elementos mecánicos que permiten desarrollar un elemento tecnológico: estructuras y mecanismos Actúa de forma dialogante y responsable en el trabajo en equipo Adopta actitudes favorables a la resolución de problemas técnico desarrollando interés y curiosidad hacia la actividad tecnológica Analizar y diseñar circuitos eléctricos en continua Señala características básicas y aplicación de componentes: resistores fijos, variables, pulsadores, motor, LEDs Distinguir aspectos básicos de la programación de sistemas electrónicos digitales. Desarrollar, en colaboración con sus compañeros de equipo, un proyecto de sistema robótico Montaje del circuito y su explicación. Conexionado a la placa Arduino UNO. Informe del trabajo realizado. Analizar circuitos en CC con la Ley de Ohm. Distinguir significado de circuito abierto o cortocircuito. Conocer las magnitudes de los circuitos. Señalar y analizar los diferentes componentes electrónicos de los circuitos. Identificar el patillaje de los distintos componentes electrónicos. Realizar el montaje de acuerdo con un esquema. Describir la conversión analógico - digital y viceversa.

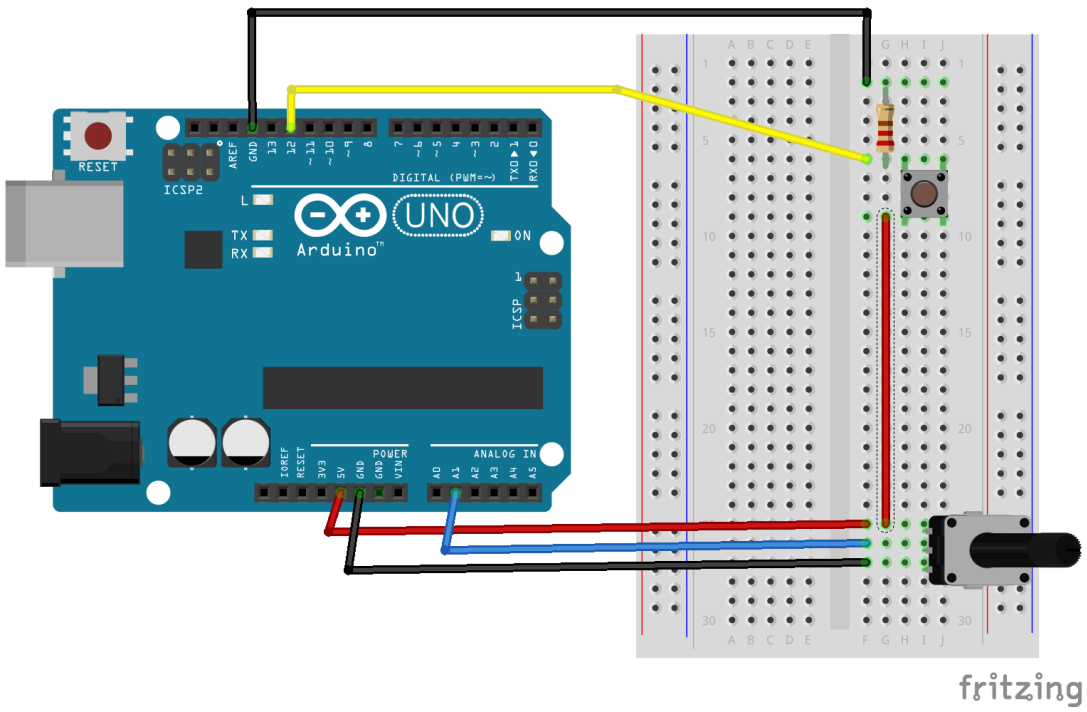
9 Código del programa de Arduino

En línea de comandos, dentro de la carpeta *s2a_fm-master,* hay que escribir el comando python s2a_fm.py COM3 (en mi caso Arduino usa el puerto COM3), para que el s2a funcione y se pueda conectar Scratch 2.0 con Arduino

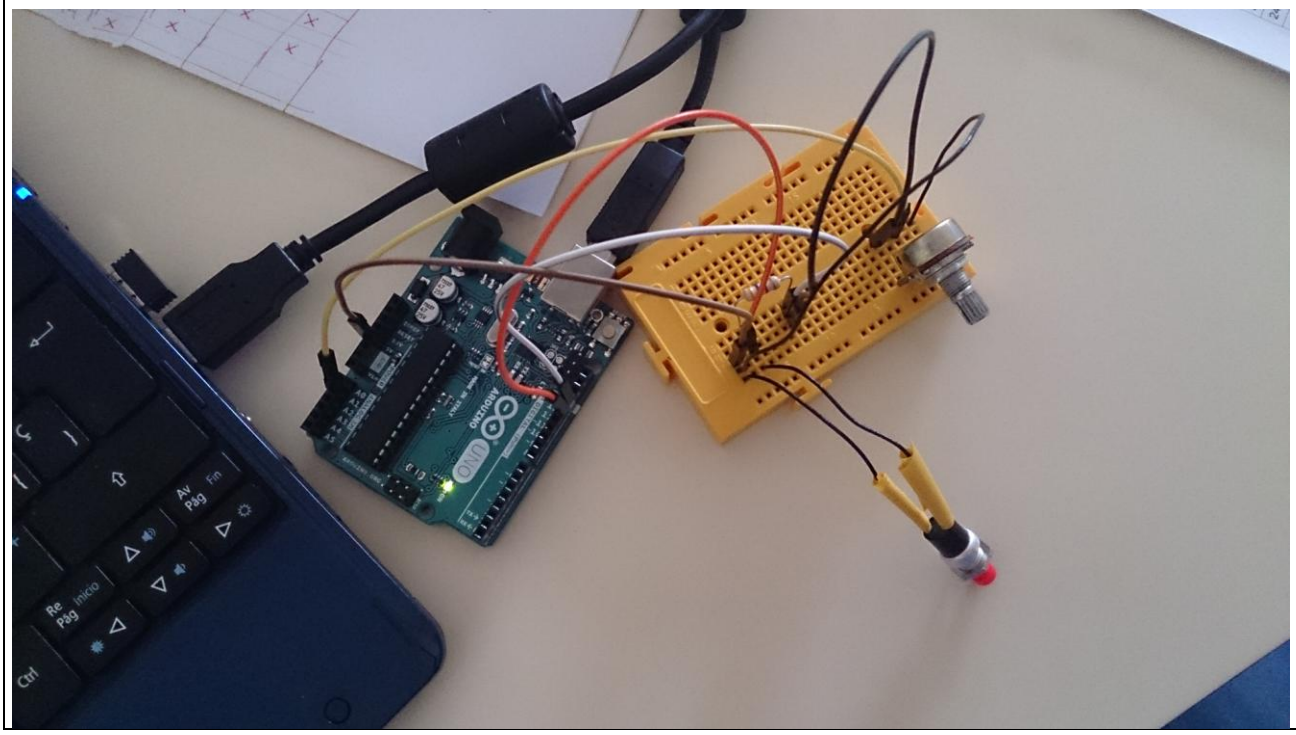
En Scratch:

Al igual que en Arduino, lo primero es activar las entradas qeu se van a usar, la digital para el pulsador, la análogica para el potenciómetro. Defino una variable, disparo, que almacena continuamente el valor de la lectura del Pin digital, 0 si no está pulsado, 1 si está pulsado. Y cuando ese valor sea 1 (es decir, se pulse el botón), se crea un clon de la flecha y se dispara. Le he dado un pequeño retardo para evitar el efecto metralleta. Por otro lado, defino otra variable "Ángulo de tiro", a la que adjudicamos el valor que se lea del pin análogico. Para mejorar su comportamiento se divide ese valor entre 1024 (que son los que puede tomar una variable análogica), lo múltiplico por 150 (que serán los grados que se va a mover el arco, y le resto 35 para ajustar la dirección de tiro. Además, se redondea el resultado final.

10 Esquema de conexionado



11 Fotografía del montaje



X Marque con una cruz si acepta que esta unidad didáctica quede recogida bajo una licencia Creative Commons (by – nc – sa) (Reconocimiento, No Comercial, Compartir Igual) para ser compartida por todos los miembros de la comunidad de docentes: http://es.creativecommons.org/blog/licencias/

Anexo: Instalación de s2a_fm desde cero y sin conocimientos de Python

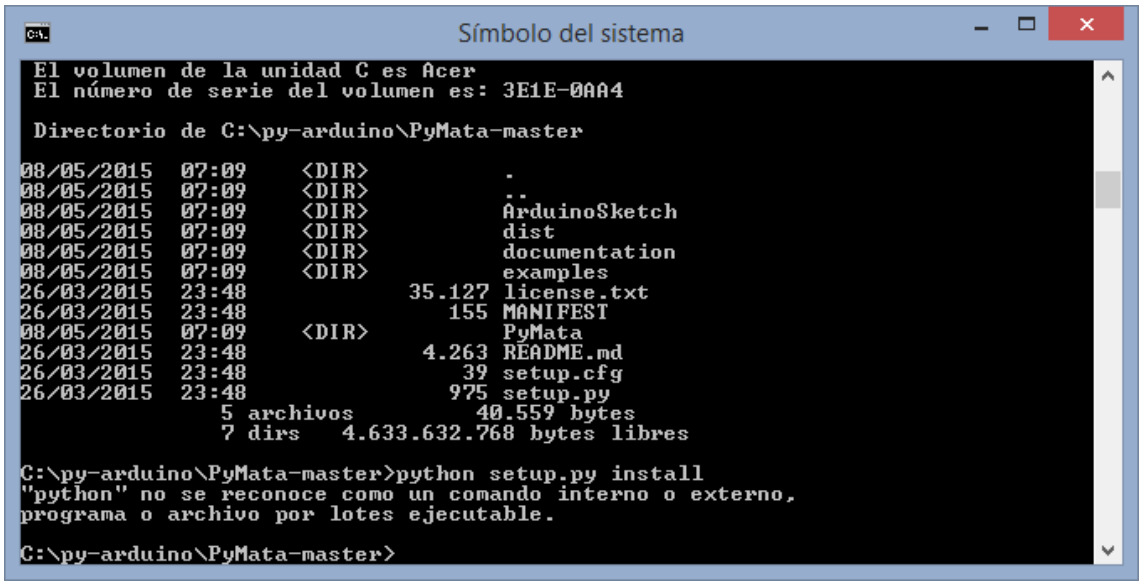
Conseguir que nuestro Scratch 2.0 tenga los bloques para poder controlar la placa Arduino no es sencillo. Existe una alternativa a s4a (que usa la versión 1.4 de Scratch), que es s2a_fm, construida sobre Python. Instalarla no es trivial. He seguido las instrucciones de https://github.com/MrYsLab/s2a_fm/blob/master/documentation/Espa%C3%B1ol/s2a_fm_Espanish_tutorial.pdf

y lo he descargado de:

https://github.com/MrYsLab/s2a_fm

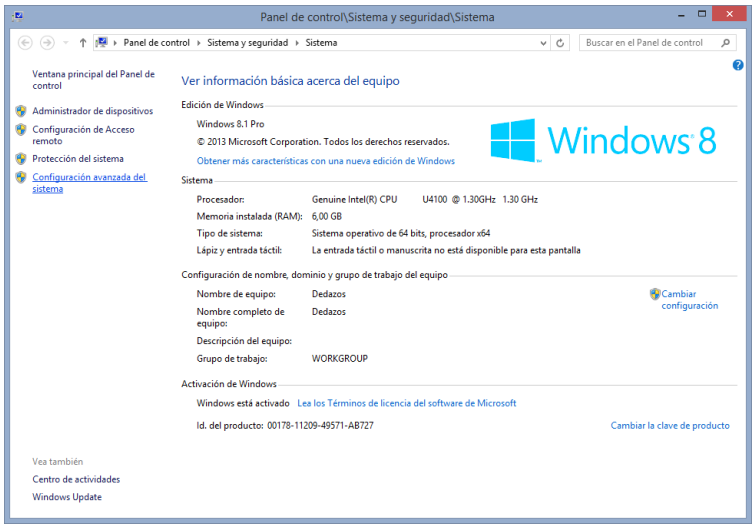
Los pasos son:

- se instala python. ¡Cuidado! Hay que instalar la versión 2.7, con la 3.x no funciona.
- se instala pyserial
- se instala pymata. Este es un rollo, porque no tiene autoinstalador. Se supone que hay que dar una orden: *python setup.py install* dentro de la carpeta donde hemos descomprimido el pymata.
  pero la línea de comandos de windows no me lo reconoce, y desde python no sé ni parece que sea el camino.

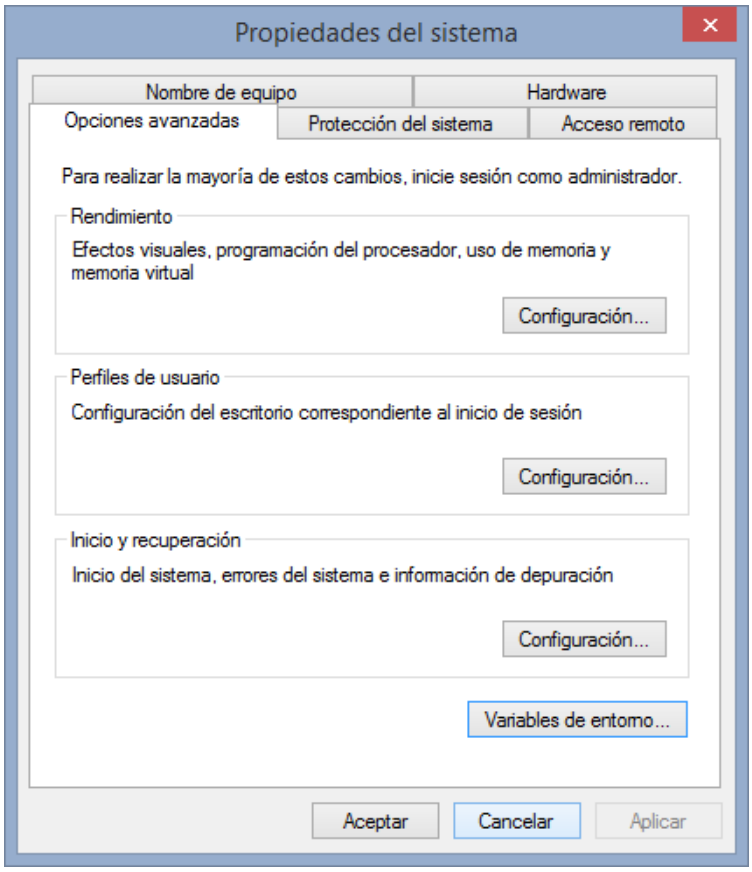


Recuerdo que para abrir la línea de comandos, pincho el botón de windows y escribo **cmd**. Una vez allí, con la orden ***cd..*** subo un nivel de directorio; con cd y el nombre de carpeta nos introducimos en ese nivel; con **dir** nos lista todos los archivos y carpetas de ese directorio; y si escribo ***cd*** y luego voy pulsando la **tecla tabulador**, van escribiendose sucesivamente todas las carpetas del directorio. De esa manera, nos ahorramos escribirlo. Si queremos retroceder en el orden de carpetas pulsamos ***Mayúsculas + tabulador***

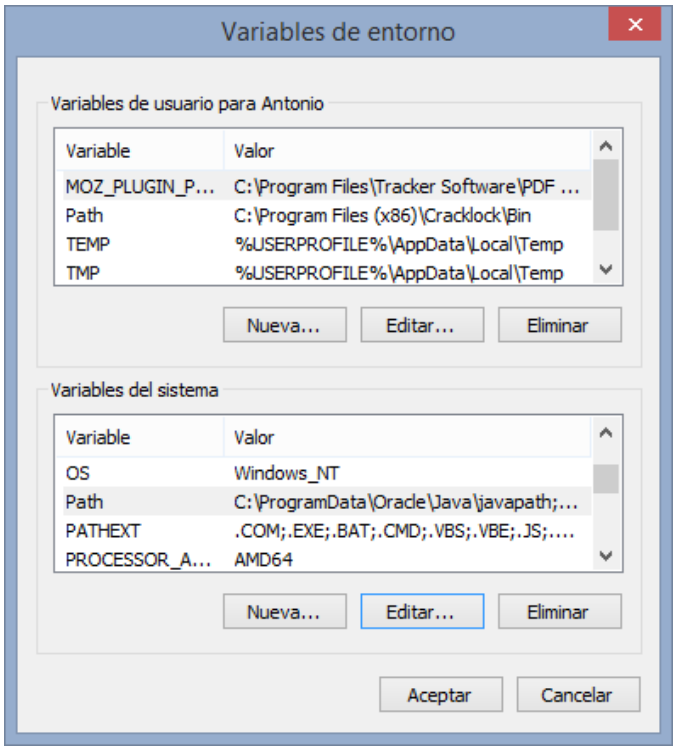
Para arreglar ese error, hay que hacer que la línea de comandos de windows, cmd, entienda las órdenes de Python. Para ello, se añade en el path de las variables de entorno el de python (se añade con un ";", para que valgan los 2) Para ello, en panel de control/ sistema, vamos a configuración avanzada del sistema:



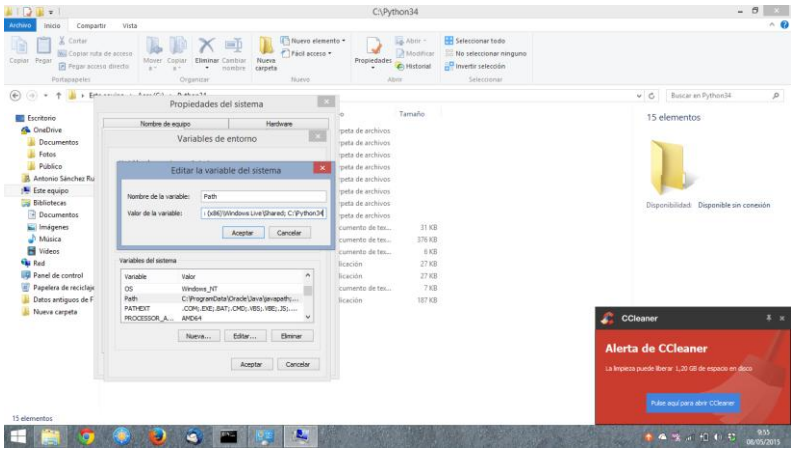
aquí pinchamos en variables del entorno



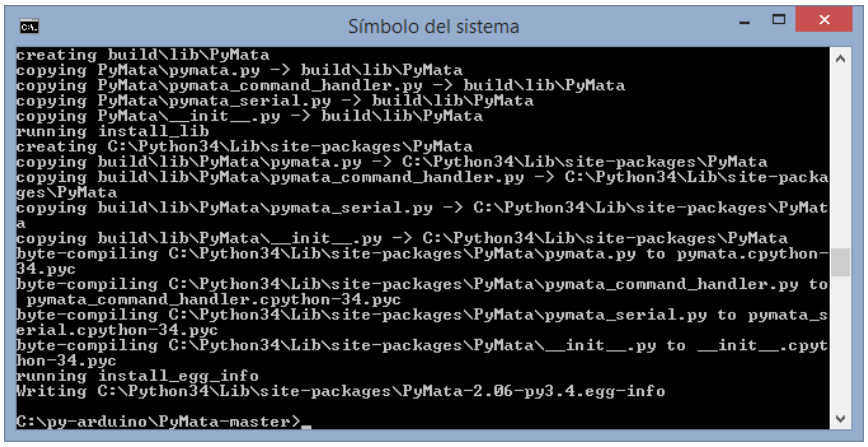
En las variables de sistemaa buscamos PATH



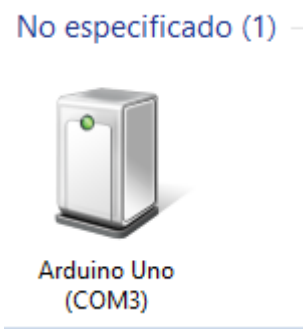
y editamos, añadiendo con un ; la ruta donde está instalado Python (en este caso c:\python27)



Se supone que ahora el cmd reconoce a Python, y por tanto instala el pymata con la orden *python setup.py install*:



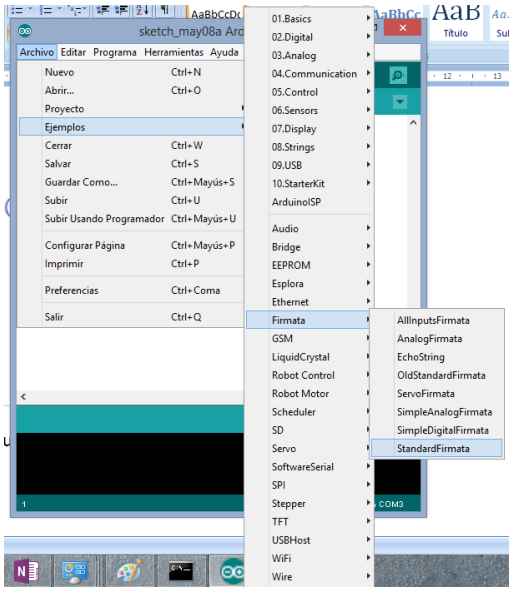
Una vez hecho esto, conecto el arduino y en panel de control/dispositivos e impresoras, compruebo en que puerto se conecta:

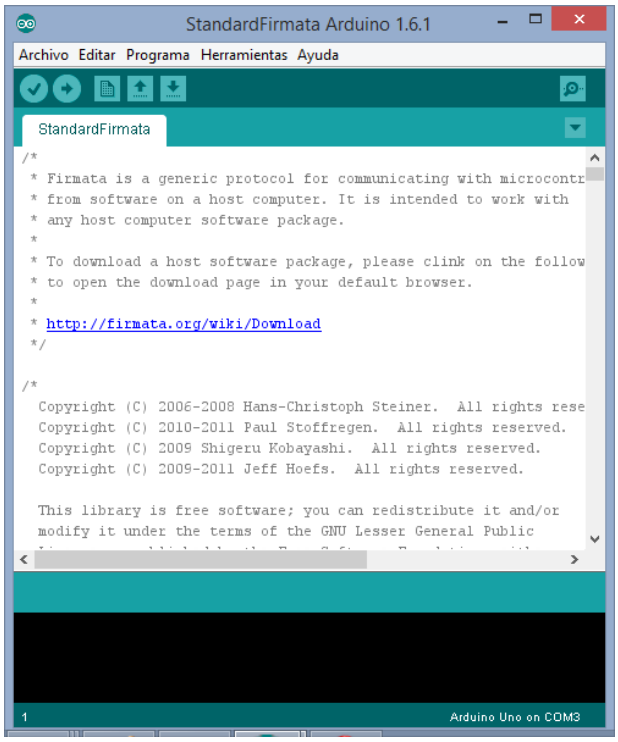


veo que esta en el puerto com3

Abro el programa de Arduino

En Archivo/Ejemplos/Firmata, Abro standard firmata:



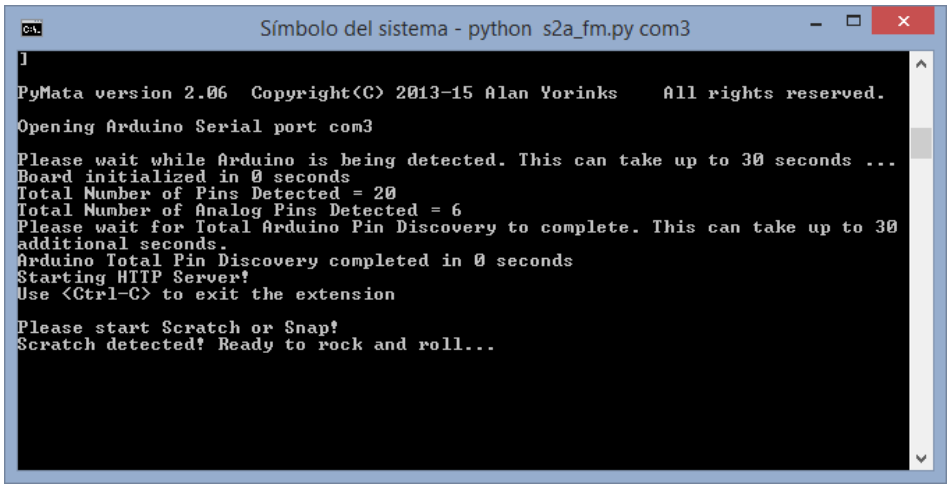


y lo mando a la tarjeta

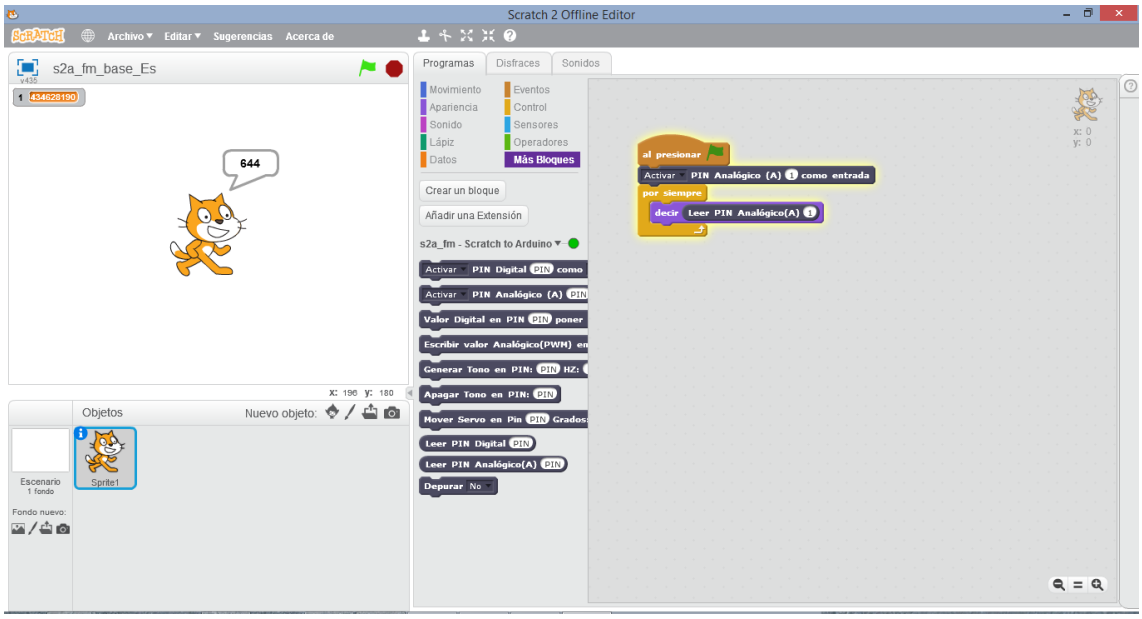
Es necesario que esté cargado para que s2a_fm funcione y conecte la tarjeta con Scratch. En caso de que la utilicemos para otros programas, tendríamos que volver a cargarlo.
Ahora navego en el cmd hasta la carpeta s2a_fm_master y escribo:

python s2a_fm.py COM3

¡Funciona!



Si ahora abro Scratch, y dentro de él abro el archivo de ejemplo *s2a_fm_base_Es.sb2*, que está en la carpeta s2a_fm_MASTER/ ScratchFiles/ScratchProjects, aparecen los bloques negros, que son los que permiten controlar la placa Arduino



Como he comentado anteriormente, el fallo es que no he sido capaz de añadir estos bloques a un proyecto guardado con anterioridad. Es decir, hay que empezar los proyectos desde este arvchivo para que ya los tenga.

En algún caso de programas poco complicados, podemos ir pinchando uno a uno los sprites y fondos, y darle al botón derecho/salvar en disco local. De esa manera se nos copian cada sprite o fondo con sus disfraces y bloques de programa. Habría que añadir las variables, y no sé si alguna otra cosa que permiteira "reconstruir" el archivo completo para poder aprovechar los archivos de la página de Scratch o los que ya tengamos programados.

Espero que os sirva de ayuda.