

SISTEMA DIDÁCTICO PARA EL APRENDIZAJE EN LA PROGRAMACIÓN ENFOCADA NIÑOS DE PRIMARIA

Mariana Betzabeth Estrada Cortez

Instituto Tecnológico de Celaya

betza.esco@gmail.com

Teresa de Jesús Iñiguez Alonzo

Instituto Tecnológico de Celaya

teresa.iñiguez23@gmail.com

Elizabeth Espitia Romero

Instituto Tecnológico de Celaya

espitia.e@hotmail.com

Resumen

Este proyecto se orienta al diseño de una plataforma en la cual se puedan desarrollar programas con fines didácticos, interactuando entre un entorno virtual y real. Dicho proyecto fue desarrollado por medio de una comunicación serial entre el software Scratch y la tarjeta LaunchPad MSP430G2553 programado con el software Energía, una forma más fácil y sencilla de entender que en CCS (Code Composer Studio) el cual desarrollo el MIT (Institute Technology of Massachusetts). Esta comunicación permitió el intercambio de datos analógicos y digitales teniendo un control externo de los escenarios creados en el programa Scratch por medio de la tarjeta LaunchPad de Texas Instrument.

Palabras Claves: Scratch, Launchpad MSP430G2553, vinculación, Energía.

1. Introducción

Esta nueva forma de aprendizaje es un sistema interactivo con aplicación didáctica, su principal función es facilitar el aprendizaje de la programación primordialmente en los

niños ya que ofrece un entorno de programación más atractivo y accesible; de igual forma, es útil para toda aquella persona que se inicia en la programación.

El software Scratch se acopla a la tarjeta Launchpad MSP430G2553 mediante el empleo de librerías que fueron incluidas en la plataforma de desarrollo CCS (desarrolladas por el MIT). En este artículo se pretende de explicar la migración de CCS a Energía dando paso a un mejor entendimiento del programa para la recepción y emisión de datos con el software.

2. Desarrollo

Plataforma Energía

Energía es una plataforma de creación de prototipos electrónicos de código abierto, también es un entorno de Desarrollo Integrado para la placa MSP430 LaunchPad de Texas Instruments. Energía cuenta con una interfaz de programación y el mismo lenguaje de programación de Arduino (Pseudo C), los mismos ejemplos básicos y varias librerías de Arduino están presentes.

El programa para realizar la vinculación de Scratch con Launchpad se realizó a través de esta plataforma, adaptándose el código de Code Composer Studio (CCS).

Tarjeta Launchpad MSP430G2553

EL MSP430 es una familia de microcontroladores fabricados por Texas Instruments (ver figura 1). Construido con una CPU de 16 bits, el MSP430 está diseñado para aplicaciones empotradas de bajo costo, sistemas inalámbricos y/o de ultra bajo consumo de energía. Esta arquitectura tiene reminiscencias de la del procesador DEC PDP-11 es perfecto para aquellos que empiezan a adentrarse en la programación de microcontroladores.*

Algunos periféricos opcionales no tan usuales incluyen comparadores (que se utilizan con el timer para funcionar como un ADC simple), amplificadores operacionales dentro del chip para el acondicionamiento de señales, conversores DAC de 12 bits, controlador de pantalla LCD, multiplicador hardware, y DMA para el manejo de datos entre zonas de memoria y/o los periféricos.[2]



Fig. 1. Tarjeta Launchpad MSP430.

Programación

En esta etapa se trabajó en base al programa principal desarrollada por el MIT [4], adaptando el programa depurando los errores, debido a la sintaxis que usa Energía contra la versión del otro compilador aunque proceden del mismo lugar.

Primero se cambian las librerías porque las utilizadas no son requeridas en el programa energía ya que no son útiles o no son existentes, como se muestra a continuación:

CCS	Energía
#include <msp430.h>	#include<Wire.h>
#include <legaymsp430.h>	//librerias agregadas
#include "serial.h"	#include<TimerSerial.h>
#include "pin.h"	#include<intrinsics.h>
	#include "pin_msp430.c"

El resto del programa que realizó el MIT [4], fue modificado completamente ya que algunos comandos no eran necesarios y otras partes se acoplaron a la estructura de Energía, para tener un código más transparente y así también poder adaptarlo a otros sistemas. En la figura 2 se muestra una parte de corrección de código entre CCS y Energía.

Ya terminada la programación y cargado el programa al microcontrolador, se busca que estén configurados los puertos RX y TX que son utilizados para la comunicación con la pc, en la fig. 3 se muestra dónde están localizados los puertos, y serán conectados a la pc y con Scratch para entablar la comunicación y para ello se descargó un driver de PicoCricket [1], específico para el funcionamiento.

```

void loop()
{
    unsigned char cnt = 0;
    unsigned char c;

    clock_init();

    pin_reserve(PIN_1_1);
    pin_reserve(PIN_1_2);

    serial_init(BAUDRATE);

    setup();

    for(;;) {
        c = serial_recv_blocking();

        if(c == SCRATCH_DATA_REQUEST)
        {
            if(cnt++ ==
COMM_BLINK_RATE) {
                // blink COMM
                to show data was requested
                pin_toggle(COMM);
            }

            cnt = 0;

            read_samples();
        }
    }
}

xmit_samples();

return 0;

CCS

int
main(void)
{
    unsigned char cnt = 0;
    unsigned char c;
    clock_init();
    pin_reserve(PIN_1_1);
    pin_reserve(PIN_1_2);
    serial_init(BAUDRATE);
    setup();
    for(;;) {
        c = serial_recv_blocking();
        if(c == SCRATCH_DATA_REQUEST) {
            if(cnt++ == COMM_BLINK_RATE) {
                pin_toggle(COMM);
                cnt = 0;
                read_samples();
                xmit_samples();
            }
        }
    }
    return 0;
}

```

Fig. 2 Código CCS y Energía.



Fig. 3. Adaptación de TXD y RXD.

Teniendo lo anterior se procedió a realizar el control externo de la interactividad del objeto virtual por medio del Launchpad, adaptándose diferentes componentes

electrónicos y estos a su vez se enlazaron a la tarjeta. Los componentes electrónicos utilizados son los siguientes:

- Sensor infrarrojo Sharp GP2Y0A02YK0F
- Push Button
- Potenciómetros
- LED's

Lo anterior se muestra en el diagrama de la fig. 4 en donde se puede ver la conexión de los componentes con la tarjeta.

Estos dispositivos acatan las órdenes de la tarjeta y el Scratch para efectuar los diferentes tipos de movimientos del personaje, los cuales fueron previamente establecidos en el área de programación.

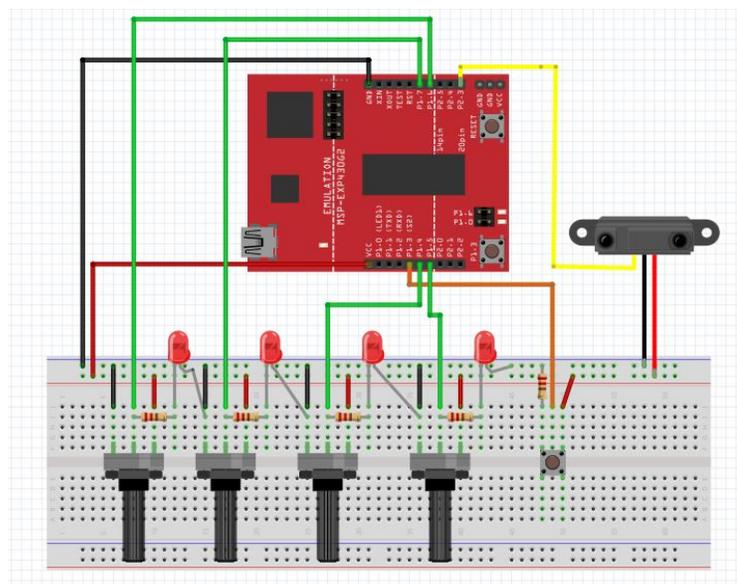


Fig 4. Diagrama de conexión.

Desarrollo escenario Scratch

Scratch es un entorno de aprendizaje desarrollado por Lifelong Kindergarten Group el MIT Media Lab in Smalltalk, utilizando Squeak. Es 100% software libre, y pretende acercar la programación a todos a partir de 8 años. Actualmente se está utilizando en

todo el mundo, y las experiencias con Scratch se encuentran en todas partes, en particular dentro del congreso más importante del mundo de Computer Science Education vinculado al grupo educación en informática de la ACM (estos últimos dos años, por ejemplo, podemos encontrar diversas referencias [5,6,7])

El lenguaje de programación vinculado al entorno Scratch es un lenguaje imperativo con las construcciones fundamentales (variables, asignación, bucles, condicionales y llamadas a funciones y acciones) ampliadas con un grupo muy numeroso de instrucciones para poder trabajar en proyectos multimedia. Un proyecto Scratch será típicamente un juego o una animación sofisticada con dibujos, iconos, sonido y otros gráficos en movimiento.*

3. Resultados

Con todo lo anterior se logró la conexión entre en software Scratch y Launchpad, logrando el control de un escenario y sus componentes con un control externo creado a partir de componentes básicos para que los niños comprendan el funcionamiento de la parte virtual y la parte física, en la fig. 5 se muestra uno de los programas realizados para la demostración de dicha vinculación.

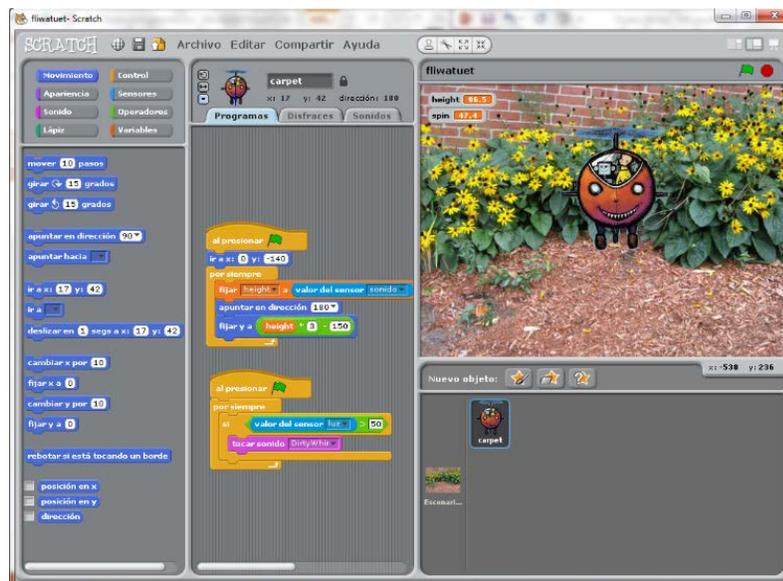


Fig. 5 Programa realizado en Scratch.

4. Discusión

Los resultados de esta investigación comprueban que es posible la migración de códigos entre lenguajes de programación, creando así un programa más entendible y corto, que fue lo que se realizó.

Así mismo se buscó la adaptación de las librerías ya existentes en Energía a las que fueron creadas en CCS al gusto del autor, así también cambiando parte del programa principal y sintaxis, para que éste sea un código transparente para Energía y Arduino.

5. Conclusiones

El trabajo obtenido fue satisfactorio ya que la adaptación del Scratch al Launchpad puede dar paso al desarrollo de nuevas investigaciones.

Se espera que este proyecto pueda sentar las bases para nuevos sistemas de aprendizaje para los centros de interacción, logrando así en un futuro el control de un robot o una serie de ellos.

6. Referencias

- [1] Company, T.P. (2006). Picocricket. Obtenido de Picocricket: <http://picocricket.com/picoboardsetupUSB.html>. Consultado 23 septiembre 2013.
- [2] Davies, J. (2008) MSP430 Microcontroller basic. 1ra edicion. 2008. Ed. Newnes. United States of America
- [3] Massachusetts, M.I (2003) Scratch. Obtenido de Scratch: <http://scratch.mit.edu/>. Consultado 23 septiembre 2013.
- [4] Wendler, S (26 de Octubre de 2012) uSherpa firmware pretending being a Scratch Sensor Board (Picoboard): <https://github.com/wendlers/usherpa-firmware/tree/master/scratch-sherpa>. Consultado 19 de agosto de 2013.
- [5] Portland, Oregon, USA, March 2008. ACM, ACM. David J. Malan and Henry H. Leitner. Scratch for budding computer scientists. In *38th SIGCSE technical symposium on Computer science*.
- [6] John Maloney, Kylie Peppler, Yasmin B. Kafai, Mitchel Resnick, and Natalie Rusk. Programming by choice: Urban youth learning programming with Scratch.

In 39th *SIGCSE technical symposium on Computer science education*, pages 367–371, Portland, Oregon, USA, March 2008.

- [7] Paolo A. G. Sivilotti and Stacey A. Laugel. Scratching the surface of advanced topics in software engineering: A workshop module for middle school students. In 39th *SIGCSE technical symposium on Computer science education*, pages 291–295.

7. Autores

Mariana Betzabeth Estrada Cortez actualmente está cursando el último semestre de la carrera de Ing. Mecatrónica en el Instituto Tecnológico de Celaya.

Elizabeth Espitia Romero actualmente está cursando el último semestre de la carrera de Ing. Mecatrónica en el Instituto Tecnológico de Celaya.

Teresa de Jesús Iñiguez Alonso actualmente está cursando el último semestre de la carrera de Ing. Mecatrónica en el Instituto Tecnológico de León.