UNIVERSIDAD CATÓLICA BOLIVIANA

Informe de Actividades

Centro de Investigación, Desarrollo e Innovación en Ingeniería Mecatrónica

(https://www.imt.ucb.edu.bo/cidimec/)

Universidad Católica Boliviana "San Pablo" – Regional La Paz

Equipos empleados	MSP432P401R LaunchPad Texas Instruments BOOSTXL-EDUMKII Educational CC3100 Wi-Fi BoosterPack
Proyecto en el que los equipos fueron utilizados	Mejora de la exposición a nuevas tecnologías de sistemas embebidos en estudiantes de pregrado
Investigador encargado	Jhon Abel Ordoñez Ingali – Ingeniería Mecatrónica – Universidad Católica Boliviana
Estudiantes investigadores	Amilkar Massy Fernandez – Ingeniería Mecatrónica – Universidad Católica Boliviana
Objetivo del proyecto	Desarrollar guías y manuales para la introducción de placas de desarrollo IoT cuyo público objetivo son estudiantes de la asignatura de Sistemas Embebidos I – carrera de Ingeniería Mecatrónica.
Breve descripción del desarrollo	Para la elaboración de documentación, inicialmente se desarrollaron aplicaciones sencillas que permitieron la familiaridad con los dispositivos – principalmente aquellos ejemplos con los que se proporciona los dispositivos. En ese sentido, como principal plataforma de desarrollo se empleó Code Composer Studio (CCS) de Texas Instruments y un enfoque didáctico, puesto que la documentación desarrollada tiene como fin la preparación de materiales curriculares para su posterior implementación. La planificación de avance se centró en el desarrollo de 15 laboratorios o experiencias prácticas desglosados acorde al siguiente esquema: ● Los primeros cinco laboratorios emplean la MSP432P41R LaunchPad y algunos componentes electrónicos introduciendo el concepto de GPIO, ADC, PWM, Timers, UART, así como la programación de estos elementos en CCS. ● Los siguientes siete laboratorios emplean también emplean la MSP432P41R LaunchPad, pero adicionan BoosterPack MKII aprovechando la instrumentación y sensores inteligentes que proporciona. Por tal razón, en esta sección se presenta el uso de interrupciones y algunos protocolos de comunicación como I2C y SPI. ● Los últimos tres laboratorios se emplean tanto MSP432P41R LaunchPad como la placa BoosterPack MKII y CC3100

	BoosterPack con el propósito de familiarizar el control Wifi y, por tanto, de las prácticas IoT. Consecuentemente, la documentación elaborada también consta de 15 secciones, cada una referente a cada laboratorio, pero también de un documento de introducción, cuyo énfasis es el de la instalación del entorno de desarrollo.
Justificación	Considerando que en la actualidad es de suma importancia familiarizarse con aplicaciones de IoT – que incluyen la automatización, hogares inteligentes, entre otros – es necesaria la introducción natural de estos contextos en el avance curricular académico. Ciertamente, para este propósito existen múltiples alternativas, pero no todas ofrecen la exposición a la tecnología ARM Cortex. La razón de este marcado énfasis sobre ARM radica en las estimaciones mayoritarias que la empresa tiene como cuota en el mercado de dispositivos móviles. La tecnología ARM es sin duda importante de analizar y estudiar durante una carrera tecnológica como lo es Ingeniería Mecatrónica. No obstante, la falta de recursos en español y documentación de inicio sencilla de entender constituyen las principales dificultades que los estudiantes encuentran al momento de desarrollar autónomamente. En efecto, el 2018 la consultora *Education First* evaluó el dominio del idioma inglés en 88 países resultando Bolivia en el puesto 61 con 48.87 sobre 100, muy debajo de otros países de Latinoamérica como Argentina, que quedó en el puesto 27. Por tal razón, es necesaria la elaboración de documentación – a manera de andamiaje educativo – que permita la introducción y establecimiento de fundamentos de esta tecnología en español. Complementariamente, la enseñanza de sistemas embebidos frecuentemente se ve obstaculizada por dos aspectos: la predominancia del inglés en la literatura, y, en relativo grado, la escasa exposición a la práctica durante el avance teórico, es decir, índices reducidos de aprendizaje centrado en la experiencia (*learning by doing*) o en el estudiante (*student centered learning*). En consecuencia, cuando se combinan tanto la importancia de ARM y tecnologías emergentes como IoT dentro del avance curricular de sistemas embebidos, es importante abordar los temas elaborando documentación disponible en el idioma nativo y priorizando la didáctica centrada en la experiencia.
Observaciones	La documentación que fue elaborada hasta el momento se encuentra en constante evolución dado que herramientas satélites, como MQTT, bases de datos, servidores para IoT, y esquemas de desarrollo, también se encuentran en constante evolución.

Código Ejemplo

```
#include "msp.h"
#include <stdint.h>


void PORT1_IRQHandler(void)
{
  // LED1
  if ( ( P1->IFG & BIT1 ) == BIT1 )
  {
          P1->OUT  ^=          BIT0;
          P1->IFG   &=         ~BIT1;
  }

  // RGB LED2
  if ( ( P1->IFG & BIT4 ) == BIT4 )
  {
          switch ( cledRGB ){
    case  1:
          P2->OUT  &=          ~( BIT0 | BIT1 | BIT2 );
      P2->OUT      |=          BIT0;
      cledRGB      =           2;
      break;

    case  2:
          P2->OUT  &=          ~( BIT0 | BIT1 | BIT2 );
      P2->OUT      |=          BIT1;
      cledRGB      =           3;
      break;

    case  3:
          P2->OUT  &=          ~( BIT0 | BIT1 | BIT2 );
          P2->OUT  |=          BIT2;
          cledRGB   =          1;
          break;

    default:
          P2->OUT  &=          ~( BIT0 | BIT1 | BIT2 );
          cledRGB   =          1;
          break;
          }

          P1->IFG   &=         ~BIT4;

  }
```

```
  NVIC_DisableIRQ  ( PORT1_IRQn );
  TIMER_A0->CTL   |=          TIMER_A_CTL_MC_1;                    //
TimerA0 ON
}

void TA0_N_IRQHandler(void)
{
            switch ( TIMER_A0->IV ){
                        case        0x00:
                                    break;

                        case        0x02:
                                    break;

                        case        0x04:
                                    break;

                        case        0x06:
                                    break;

                        case        0x08:
                                    break;

                        case        0x0A:
                                    break;

                        case        0x0C:
                                    break;

                        case        0x0E:

                                    TIMER_A0->CTL       &=
~TIMER_A_CTL_IFG;
                                    break;

                        default:
                                    break;
            }



            if ( ( ( P1->IFG & BIT4 ) == BIT4 ) || ( ( P1->IFG & BIT1 ) == BIT1 )
)
```

```
            {
                        P1->IFG   &=          ~( BIT1 | BIT4 );
            }
            else
            {
                        TIMER_A0->CTL       &=
~TIMER_A_CTL_MC_1;
                        NVIC_EnableIRQ      ( PORT1_IRQn );
            }
}

void main(void)
{
            // Unlock CS module for register access
            CS->KEY = CS_KEY_VAL;
            CS->CTL1 |= CS_CTL1_SELA_1;
            CS->KEY = 0;

            // Stop watchdog timer
            WDTCTL = WDTPW + WDTHOLD;

            //Configuracion de puertos
            P1->OUT  = 0x00;     P1->DIR  = 0xFF;
            P2->OUT  = 0x00;     P2->DIR  = 0xFF;
            P3->OUT  = 0x00;     P3->DIR  = 0xFF;
            P4->OUT  = 0x00;     P4->DIR  = 0xFF;
            P5->OUT  = 0x00;     P5->DIR  = 0xFF;
            P6->OUT  = 0x00;     P6->DIR  = 0xFF;
            P7->OUT  = 0x00;     P7->DIR  = 0xFF;
            P8->OUT  = 0x00;     P8->DIR  = 0xFF;
            P9->OUT  = 0x00;     P9->DIR  = 0xFF;
            P10->OUT = 0x00;    P10->DIR = 0xFF;
            PJ->OUT    = 0x00;    PJ->DIR     = 0xFF;
            conf_IO    ();
            conf_TA                ();

            //botones msp432
            P1->DIR    &=          ~( BIT1 | BIT4 );
            P1->OUT    |=           ( BIT1 | BIT4 );
            P1->REN    |=           ( BIT1 | BIT4 );

            // LED1     ( Digital GIPO )
            P1->SEL0  &=          ~( BIT0 | BIT1 | BIT4 );
            P1->SEL1  &=          ~( BIT0 | BIT1 | BIT4 );
```

```
            // LED2 RGB          ( Digital GIPO )
            P2->SEL0  &=        ~( BIT0 | BIT1 | BIT2 );
            P2->SEL1  &=        ~( BIT0 | BIT1 | BIT2 );


            P1->IES   |=         ( BIT1 | BIT4 );
            P1->IFG   &=        ~( BIT1 | BIT4 );

            P1->IE    |=         ( BIT1 | BIT4 );


            NVIC_EnableIRQ                ( PORT1_IRQn );
            NVIC_SetPriority    ( PORT1_IRQn, 0 );

            //timer
            TIMER_A0->CCR[0] =             1410;


            TIMER_A0->CTL      =           TIMER_A_CTL_TASSEL_1 |
TIMER_A_CTL_CLR | TIMER_A_CTL_IE;   // ACLK, TA0 interrupt ON

            NVIC_EnableIRQ                ( TA0_N_IRQn );
            NVIC_SetPriority    ( TA0_N_IRQn, 0 );

            uint8_t   cledRGB  =          1; //1 para rojo, 2 para verde y 3
para azul
            __enable_irq();

            // Do not wake up on exit from ISR
            SCB->SCR |=         SCB_SCR_SLEEPONEXIT_Msk;

            while (1){
                        __sleep   ();
            }
}
```