

Proyecto Final
Control Remoto de una casa mediante GSM / Tivafono
Rojas Juárez Adolfo

Enlace a Repositorio: <https://github.com/AdolfoRojasJ/Tivaphone>

1. Introducción

En este reporte se explica el desarrollo realizado en el microcontrolador TIVA C Series para configurar su funcionalidad como un teléfono así como manejo remoto de ciertas cualidades de un hogar mediante mensajes SMS a este mismo.

2. Antecedentes

Los microcontroladores TIVA C Series son una serie de microcontroladores lanzados por Texas Instruments con puertos analógicos, bajo consumo eléctrico, con un procesador mononúcleo ARM Cortex M4 incluido a 125 MHz. Entre sus características se incluye memoria Flash de 256 KB, SRAM de 32 KB, una EEPROM de 2KB, 3 comparadores analógicos, interfaces seriales incluyendo 8 UART's, 4 SSI/SPI, 2 CAN y 6 I2C (Morales, M. 2013).

El módulo A6 es un componente que incluye GSM/GPRS/GPS, funciona mediante comandos AT que se envían por comunicación en serie.

3. Material

- 2 Tiva's C Series TM4C1294NCPDT.
- SIM Telcel.
- Módulo IoT A6.
- Display LCD 16x2.
- Teclado Matricial.
- Placas fenólicas.
- 2 Fuentes 5V 1A.
- 4 Leds blancos.
- Led Rojo.
- Módulo RTC DS1307.
- Matriz Led Neopixel 8x8.

- Buzzer.
- Motor a pasos con su respectivo Driver.
- Push Button.
- 2 Servomotores.
- Resistencias.
- 2 Optoacopladores 6N137.
- 2 Reguladores 7805.
- 2 Transistores IRF3205.
- 2 NTE4920.
- 2 Transistores 2N2222A.
- Fuente externa de Voltaje
- Foco incandescente.

4. Hardware

Se utilizaron 2 microcontroladores TIVA C Series conectados vía UART, se conectó un módulo IoT A6 a una de ellas para poder realizar comunicación vía mensajes de Texto.

5. Software

Se utilizó el IDE Code Composer Studio para la programación, todo se realizó en lenguaje C, se usaron las librerías externas necesarias para trabajar con el microcontrolador, las cuales se consiguieron del siguiente enlace: <https://www.ti.com/tool/SW-TM4C>.

6. Instrucciones

En esta sección se explicarán los pasos necesarios para el correcto funcionamiento del Tiváfono y la conexión entre TIVA's para tener control vía SMS.

6.1. Configuración del proyecto en Code Composer

Se explicará como configurar correctamente el proyecto en el IDE.

6.1.1. Creación de un Proyecto en Code Coposser

Se dá clic en *New* y se elige la opción *CCS Project*, se nombra el proyecto como el usuario desee, cuidando que el Target sea *TIVA TM4C1294NCPDT* y la opción Connection sea *Stellaris In-Circuit Debug Interface*, una vez se elijan estas opciones, se crea el proyecto.

6.1.2. Inclusión de librerías externas

Se da clic derecho en el proyecto y se selecciona la opción *Properties*, damos clic en *Include Options* y añadimos la ruta donde instalamos las librerías del SDK (Tivaware).

Igualmente en las opciones de *Arm Linker* damos clic en *File Search Path*, y añadimos la librería *driverlib*, la cuál se encuentra en la ruta donde descargamos Tivaware y de ahí accedemos a: *driverlib/ccs/Debug/driverlib.lib*

6.2. TIVA 1. TIVAFONO

El pimer código se basa en un teléfono, el cuál es capaz de realizar llamadas, mandar mensajes y recibir mensajes para lanzar señales de control a la segunda TIVA, a manera de resúmen se explicará en las siguientes líneas partes fundamentales del código, pero el código completo se puede visualizar en el repositorio Github.

Las librerías utilizadas en el código fueron

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h>
#include "inc/tm4c1294ncpdt.h"
#include "driverlib/rom_map.h"
#include "driverlib/sysctl.h"
#include <stdio.h>
#include <string.h>
```

Figura 1: Librerías Utilizadas

Se declararon variables y arreglos los cuáles ayudaron a almacenar datos de interés, algunas funcionaron como banderas o contadores, igualmente se declararon diversas funciones las cuales ayudaron a manejar los cambios en el Display LCD o algunas fueron para Envío y Recepción de Datos por UART. El código principal se dividió a manera que el teléfono funciona de la siguiente forma:

MODO 1 TELEFONO (A)

- OPCION 1 LLAMAR (A)
- OPCION 2 COLGAR (B)
- OPCION 3 BORRAR (C)
- OPCION 4 REGRESA A HOME (*)

MODO 2 MENSAJE (B)

- OPCION 1 ENVIAR (A)
- OPCION 3 ELIMINAR TODO EL MENSAJE (C)
- OPCION 4 REGRESA A HOME (*)

MODO 3 CONTROL (C)

- OPCION 1 ABRIR PUERTA
- OPCION 2 ENCENDER FOCO 1

- OPCION 3 ENCENDER FOCO 2
- OPCION 4 ENCENDER FOCO 3
- OPCION 5 ENCENDE FOCO 4
- OPCION 6 ACTIVAR DISPENSADOR AUTOMÁTICO DE COMIDA PARA MASCOTAS
- OPCION 7 ENCENDER MATRIZ LED
- OPCION 8 ENVIAR SEÑAL DE CONTROL DE POTENCIA
- OPCION 9 APAGAR ALARMA DE INCENDIOS

Las opciones relacionadas a realizar llamadas y mensajes se hacen enviando caracteres por UART al A6, el cual admite comandos AT, por ejemplo para realizar una llamada ocupamos el comando: ATD55xxxxxxx;

Donde las x se sustituyen por el número deseado, la lista de comandos se obtuvo de: <http://profesores.fi-b.unam.mx/use/se2024-1/gsm/comandosGSMbreviario.pdf> La función que es la inicialización de los puertos que se usaron se coloca a continuación:

```

void INI_PORTS(void)
{
    SYSCTL_RCGCGPIO_R |= 0x6A81;
    while ((SYSCTL_PRGPIO_R & 0x6A81) != 0x6A81);

    SYSCTL_RCGCUART_R |= 0x41; //UART 0 Y 6

    //PUERTO A
    GPIO_PORTA_AHB_PCTL_R = (GPIO_PORTA_AHB_PCTL_R & 0xFFFFF00) + 0x00000011;
    GPIO_PORTA_AHB_DEN_R |= 0x03;
    GPIO_PORTA_AHB_AMSEL_R &= ~0x03;
    GPIO_PORTA_AHB_AFSEL_R |= 0x03;

    //PUERTO H
    GPIO_PORTH_AHB_DATA_R = 0x00;
    GPIO_PORTH_AHB_DEN_R |= 0xFF;
    GPIO_PORTH_AHB_DIR_R = 0x0f;

    //PUERTO K
    GPIO_PORTK_DATA_R = 0x00;
    GPIO_PORTK_DEN_R |= 0xFF;
    GPIO_PORTK_DIR_R |= 0xff;

    //PUERTO M
    GPIO_PORTM_DATA_R = 0x00;
    GPIO_PORTM_DEN_R |= 0x07;
    GPIO_PORTM_DIR_R |= 0x07;

    //PUERTO P
    GPIO_PORTP_PCTL_R = (GPIO_PORTP_PCTL_R & 0xFFFFF00) + 0x00000011;
    GPIO_PORTP_DEN_R |= 0x03;
    GPIO_PORTP_AMSEL_R &= ~0x03;
    GPIO_PORTP_AFSEL_R |= 0x03;

    //PUERTO Q
    GPIO_PORTQ_DEN_R |= 0xFF;
    GPIO_PORTQ_DIR_R = 0x00;
    GPIO_PORTQ_PUR_R = 0x0f;

    //UART 0
    UART0_CTL_R &= ~0x0001;
    UART0_IBRD_R = 8;
    UART0_FBRD_R = 43;
    UART0_LCRH_R = 0x0070;
    UART0_CTL_R = 0x0301;

    //UART 6
    UART6_CTL_R &= ~0x0001;
    UART6_IBRD_R = 8; //115200
    UART6_FBRD_R = 43;
    UART6_LCRH_R = 0x0070; //8 BITS, HABILITAR FIFO
    UART6_CTL_R = 0x0301;
}

```

Figura 2: Inicialización de Puertos

Se coloca el valor 0x6A81 en el Reloj para dar a entender que usaremos los puertos A,H,K,M,P y Q, los puertos A y P se usarán para UART, los puertos K y M son usados para señales de control

del Display de 16x2 y los puertos H y Q son usados para señales del teclado Matricial, el almbrado del display y el teclado se presenta en los siguientes diagramas:

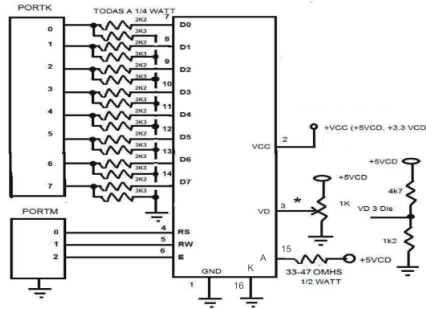


Figura 3: Circuito de conexión LCD a TIVA

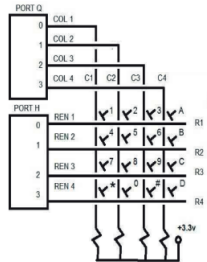


Figura 4: Circuito de conexión Teclado matricial a TIVA

En el modo control se manda a llamar a la función que lee lo recibido por UART, almacena el contenido en un arreglo, y si los elementos 73 a 76 (Ahí se almacena el contenido del mensaje) son una cadena previamente definida se le da un valor a una variable que será enviada por el otro UART a la TIVA 2.

6.3. TIVA 2. CASA

Este circuito se centrará simplemente en una TIVA que tiene las conexiones de todos los leds, servomotores, y cosas que serán controladas, el funcionamiento se limita a recibir por UART una variable con un valor preestablecido, y dependiendo ese valor, se manda a realizar la acción correspondiente, por ejemplo al recibir en la variable de dato un 31 se encenderá el LED 1 de la Casa, todo se controla directamente a través de UART; excepto la alarma de incendios, la cuál se activa debido a un ADC, si detecta temperatura arriba de 50 grados centígrados se activará el motor a pasos que simula el aspersor, aunque por SMS si será posible detenerla.

7. Conclusión

El proyecto presenta ciertas limitaciones que debido a tiempo no fueron controladas, por ejemplo, el teléfono es capaz de realizar llamadas y enviar mensajes, sin embargo no puede contestar una llamada que le entre, así como no puede revisar mensajes que recibe, sin embargo la comunicación entre ambos procesadores se realiza de manera correcta, una Tiva obtiene un dato que se envía a la otra mediante UART.

8. Referencias

- Miguel M. (2013). *An Introduction to the Tiva C Series Platform of Microcontrollers* (). Recuperado 2 de Diciembre de 2023, de <https://www.ti.com.cn/cn/lit/wp/spmy010/spmy010.pdf>
- Texas Instruments. (2014). Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit EKTM4C1294XL User's Guide (SPMU365B).
- I2C Info: A Two-wire Serial Protocol. I2C info - I2C bus, interface and protocol, 2020. Recuperado 14 de Octubre de 2023, de <https://i2c.info/>