

Proyecto Sistemas Empotrados

Autor: Agustín Jofre Millet
Curso 2018 / 2019

PROYECTO

Resumen

La idea de este proyecto es simular un sistema de domótica para el hogar. Cuenta con un sistema de seguridad , el cual permite abrir la puerta o cerrarla , e identificar a intrusos. Una vez dentro tenemos la posibilidad de escuchar música o consultar la temperatura. En total se han implementado 6 periféricos , pero solo se han usado 5. A continuación se explica en detalle cada uno de estos estados en los que puede estar nuestro sistema.

Sistema de seguridad

El sistema de seguridad consta de un motor el cual se utiliza para abrir y cerrar la cerradura de casa. Este se activa por medio del keypad, el cual se usa para introducir la clave de acceso al hogar, (Figura 2) para esto se utiliza la tecla 'A' (Acceso) seguida del código y a continuación la letra 'C' (confirmar), el mismo procedimiento se utiliza para bloquear la cerradura al abandonar la casa(Figura 1). Este código se compara con una variable global . En el caso de abrir el motor gira en sentido contrario a la agujas del reloj y al cerrar el motor gira en sentido de agujas del reloj (Figura 3.1). El cliente tiene 3 intentos para accede a la casa, en caso de superarlos suena una alarma.

Para cada uno de esos sucesos , se muestra información por la matriz de puntos (Banner) informando al cliente de los estados del sistema.



Figura 1. Sistema bloqueado.



Figura 2. Introducción de código.

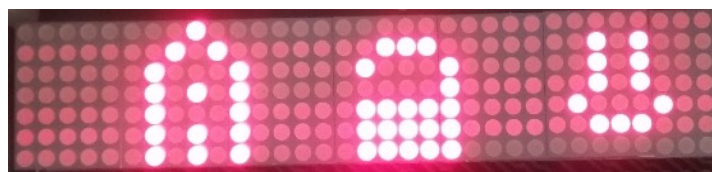


Figura 3. Acceso correcto..

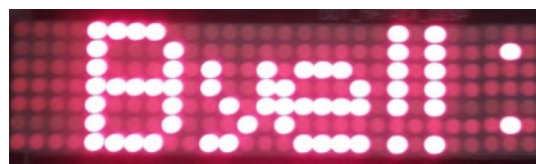


Figura 3.1 Cierre.

Reproductor de música.

Una vez dentro del hogar , podemos acceder al sistema musical , este se accede a través del teclado pulsando la letra 'E'. Una vez dentro se muestra por él banner (Figura 4) los temas que tenemos disponibles para reproducir, el cliente puede reproducir cada uno de ellos pulsando la tecla adecuada en teclado, el sonido se reproduce por medio de la salida de sonido de la FPGA, conectada al altavoz del entrenador de los laboratorios. Para salir del reproductor se debe pulsar la tecla 'D' (stop) . Una vez fuera podemos volver a entrar al reproductor o consultar otras tareas.

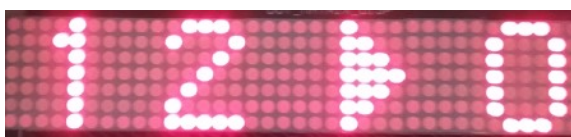


Figura 4. Opciones de reproducción.

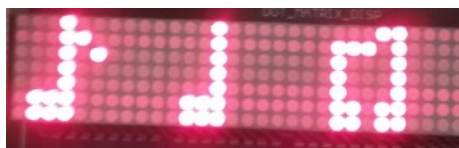


Figura 5. Reproduciendo.

Consultar temperatura.

Otra de la opciones disponible con las que cuenta el cliente , es consultar la tempera ambiente , esto se realiza a través del comando 'B' del keypad , una vez presionado se muestra por la matriz de puntos durante unos segundos la temperatura (Figura 6), para conseguir dicho dato se ha utilizado el sensor de temperatura de la placa, el cual esta conectado por medio del bus I2C, en donde se realiza la conversion analógico digital,

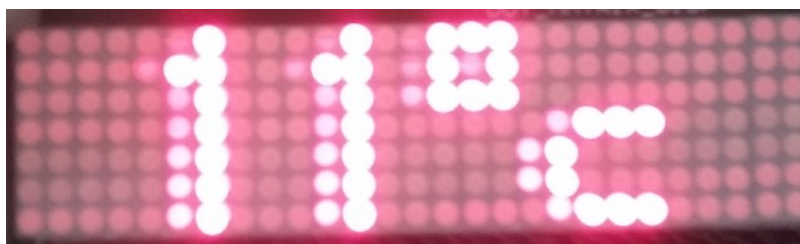


Figura 6. Temperatura.

ALGUNAS FUNCIONES EN C.

Imprimir por el banner.

Se han creado varias funciones auxiliares que imprimen distintos mensajes por el banner, una de ellas es la pantalla principal de inicio (Figura 7). En general son todas muy similares, salvo por la de imprimir temperatura ya que recibimos un valor obtenido por el bus I2C, este número hay que separarlo en dos, por lo que se imprime su cociente y su resto, seguido de 'o' y 'c'. Para codificar cada letra se ha utilizado un array bidimensional para cada caso.

```
void bannerPrintHome(){
    int i, j;

    for(j=0; j < 5; j++){
        for(i=0; i<7; ++i)
            bannerWrite(i, j, home[j][i]);
    }
}
```

```
void bannerPrintTemperatura(Xuint32 temp){
    Xuint32 coc, resto;
    coc = temp / 10;
    resto = temp % 10;
    int i;

    for(i=0; i<7; ++i)
        bannerWrite(i, 0, temperatura[coc][i]);
    for(i=0; i<7; ++i)
        bannerWrite(i, 1, temperatura[resto][i]);
    for(i=0; i<7; ++i)
        bannerWrite(i, 2, temperatura[10][i]);
    for(i=0; i<7; ++i)
        bannerWrite(i, 3, temperatura[11][i]);
}
```

Pedir Código.

Esta función va almacenando dígitos, para crear códigos de más de un dígito leído por keypad (solo dígitos). Solo se sale de la función, como ya mencionamos arriba, pulsando la tecla 'C' (confirmar).

```
Xuint32 pedirCodigo(){
    Xuint32 codeUser = 0x0, codeUserResult = 0;

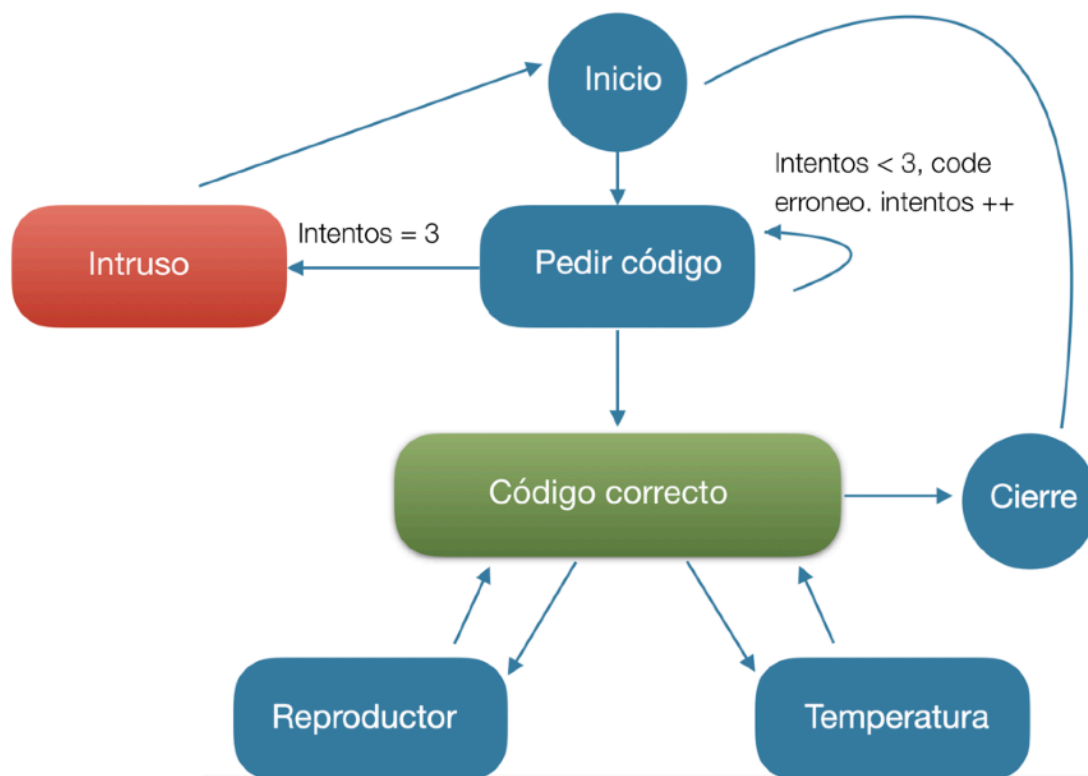
    while (codeUser != 0xC){ //confirmar code
        codeUser = keyPadRead();
        if(codeUser <= 0x9){
            codeUserResult *= 10;
            codeUserResult += codeUser;
            xil_printf("codeUserResult: %d\n", codeUserResult);
        }
    }
}
```

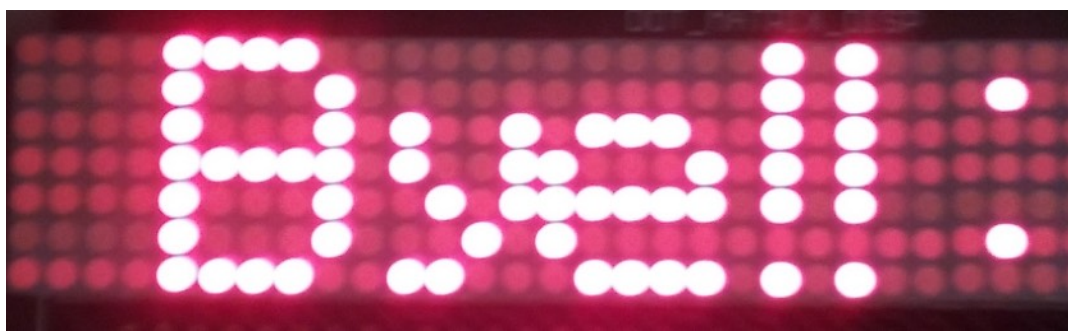
Control del motor.

Para controlar el motor necesitamos saber la dirección (reg0 (0)) , halfstep (reg0 (1)) y el numero de pasos a girar (reg0 (4 to 7). Con estos parámetros , escribimos en el motor , el cual se quedara esperando hasta terminar de girar.

```
void controlMotor(Xuint32 direccion,Xuint32 halfstep,Xuint32 numPasos){
    Xuint32 Data;
    Data = 0 ;
    Data= ((direccion << 31) | (halfstep << 29) | (numPasos << 25));
    MOTOR_HW_mWriteReg ( MOTOR_BASE_ADDR_0 , 0, Data );
    Data = MOTOR_HW_mReadReg ( MOTOR_BASE_ADDR_0 , 0); // saber si el motor esta girando o no
    while (!( Data & 0x40000000 )){
        Data = MOTOR_HW_mReadReg ( MOTOR_BASE_ADDR_0 , 0);
    }
}
```

DIAGRAMA DE FUNCIONAMIENTO.





Agustín Jofre Millet
