

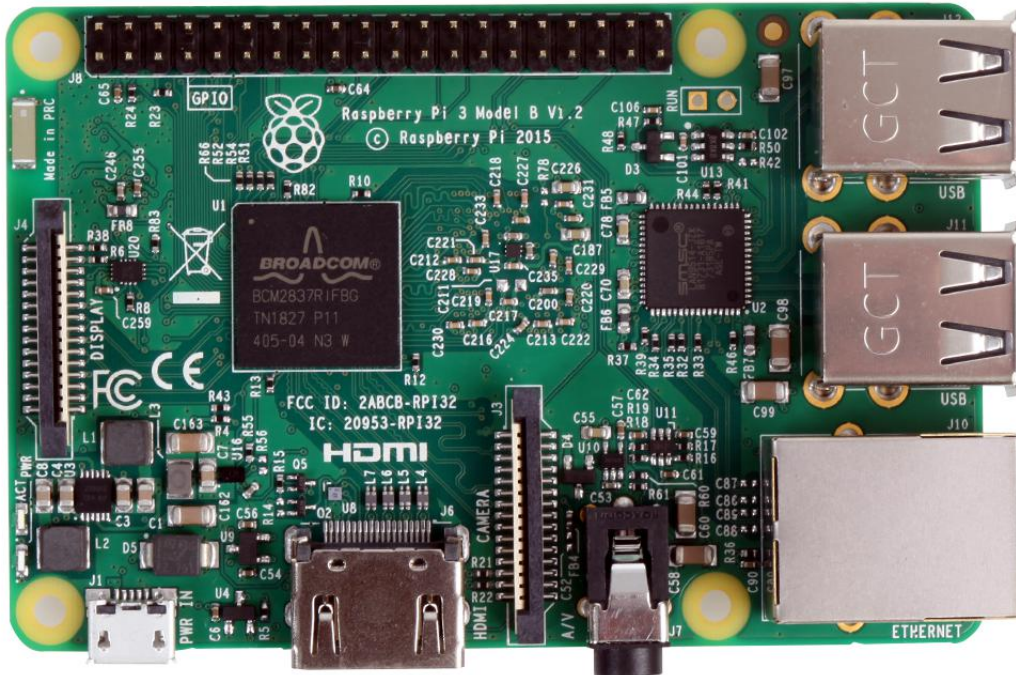
Tecnicas Digitales II
Proyecto Final
4R2

Campetella, Nazareno 78006
Vera Causich, Mariano 66878
Rosso, Guido 67756

February 8, 2021

Introducción

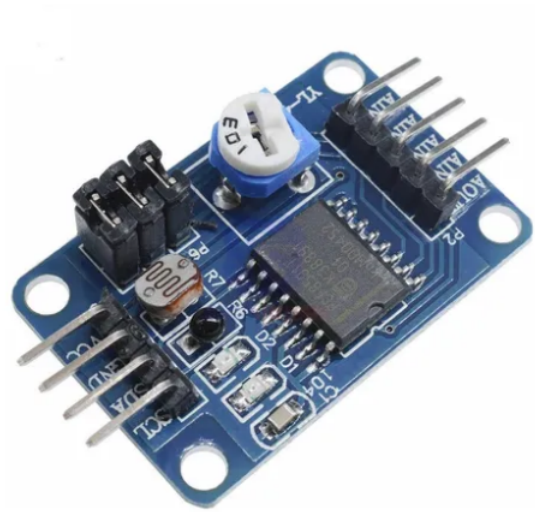
En el siguiente informe explayaremos el proyecto final basado en programacion C en placa Raspberry. Para este proyecto utilizamos el modelo Raspberry Pi 3 B y su sistema operativo Raspbian (Devian para Raspberry). Este modelo añade conexión LAN inalámbrica y Bluetooth, tiene un procesador Boardcom BCM 2937 de 64 bit Quad Core y set de instrucciones ARMv7 tambien memoria RAM de 1GB a 400MHz. Lo cual la hace un poderoso computador de pequeño tamaño e infinidad de usos.



Tambien utilizaremos una placa adicional desarrollada por la catedra y hecha por nosotros, esta nos servira para demostrar la mayoría de los programas realizados.



Aparte realizamos la compra de un modulo conversor Pcf8591 que lo utilizaremos para seleccionar la velocidad inicial de cada secuencia.



Con respecto a la comunicación en serie usamos el adaptador conversor de usb a serie ttl cp2102.



Desarrollo

Menu

Empezaremos el desarrollo mostrando el Menu principal de nuestro programa y todas sus opciones. De este iremos derivando todos y cada uno de los programas usados para ver más a fondo la codificación de estos.

```
//incluimos todos los programas externos a Menu
#include "2Passw.c"
#include "3ConfiguracionIO.c"
#include "4Autofantastico.c"
#include "5Carrera.c"
#include "6Apilada.c"
#include "7Choque.c"
#include "8Sec1_cortina.c"
#include "9Sec2_gusano.c"
#include "10Sec3_barrido.c"
#include "11Sec4_navidad.c"
#include "12Contador.c"
#include "13Comuraspi.c"
//14SeriePC.c va en la computadora
#include "15Seteoveloci.c"

//Incluimos librerias usadas poor Menu
#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"
#include <wiringPi.h>

//Inicializacion de variables
int led[8]={23,24,25,12,16,20,21,26},pulsadores[2]={17,27},llaves[4]={5,6,13,19}, contador=50000,m,psw=0,serial_port;

int main() {

    int s,b=0,aux;
    Password(); //Llamado a funcion Password en passw.c
    if(psw==1){ //Si la contrasenia es correcta retorna 1
```

Configuracion(); //Llamado a funcion Configuracion de entrada/salida en configuracionIO.c

while(1){ //Menu principal

system("clear");

if(b==0) {

printf ("BIENVENIDO AL SISTEMA\n");

b=1;

}

printf ("Funciones de luces:\n\n1-Auto Fantastico\n2-Carrera\n3-Apilada\n4-Choque\n5-Cortina\n6-Gusano\n7-Barrido\n8-Navidad\n\nOtras funciones:\n\n9-Comunicacion Serie\n10-Cambio de velocidad\n\n11-Salir\n");

scanf("%d",&s);

getchar();

//Lector de opcion elegida

switch(s){

case 1:{

aux=contador; //Variable auxiliar para respaldar la variable original de contador

AutoFantastico();

contador=aux; //Respaldo

break;

}

case 2:{

aux=contador;

Carrera();

contador=aux;

break;

}

case 3:{

aux=contador;

Apilada();

contador=aux;

break;

}

case 4:{

aux=contador;

Choque();

contador=aux;

break;

}

case 5:{

aux=contador;

Cortina();

contador=aux;

break;

```

    }

    case 6:{

        aux=contador;
        Gusano();
        contador=aux;
        break;
    }

    case 7:{

        aux=contador;
        Barrido();
        contador=aux;
        break;
    }

    case 8:{

        aux=contador;
        Navidad();
        contador=aux;
        break;
    }

    case 9: {

        Comunicacion_serie(); //Llamado a comunicacion serie entre raspberry y PC
        break;
    } Conclusión

    case 10: {

        Velocidad_inicial(); //Configuracion de velocidad inicial
        break;
    }

    case 11: {

        return 0;
    }

    default:{

        system("clear");
        printf ("\nSeleccion no valida\n\n");
        delayMillis(1000);
        break;

    }

}

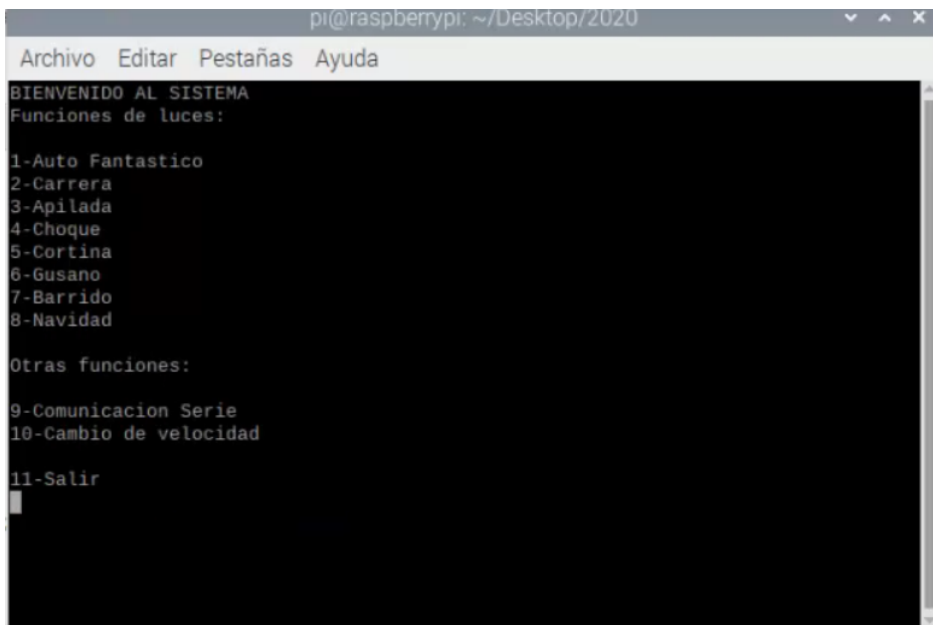
}

}

return 0;

```


}



A screenshot of a terminal window titled "pi@raspberrypi: ~/Desktop/2020". The window has a menu bar with "Archivo", "Editar", "Pestañas", and "Ayuda". The terminal content is as follows:

```
BIENVENIDO AL SISTEMA
Funciones de luces:

1-Auto Fantastico
2-Carrera
3-Apilada
4-Choque
5-Cortina
6-Gusano
7-Barrido
8-Navidad

Otras funciones:

9-Comunicacion Serie
10-Cambio de velocidad

11-Salir
```

A cursor is visible on the line "11-Salir".

Contraseña

Al ejecutar nuestro programa Menu lo primero que se ejecutara sera el programa Password el cual nos pedira la contraseña del sistema y nos dejara acceder a este hasta que la coloquemos correctamente, de no ser posible tres intentos el programa se cerrara. Para esto utilizamos la libreria curses que necesitamos instalar en nuestro raspbian A continuación dejaremos la codificación del programa Password con sus respectivos comentarios.

```
#include <curses.h>
#include <stdlib.h>
#define tam 6 //Tamanio de constrasenia
#define intro 10 //valor ASCII de enter
extern int psw; //Flag de Menu
void Password(void){
    char password[tam];
    char contra[]="bokee";
    int pos,p,ch,u,i,k=0;

    initscr(); //Iniciamos el modo Ncurses
    noecho(); //Iniciamos modo que permite Ncurses, que tiene la funcion no mostrar lo que ingresa el usuario
    for(u=0;u<3;u++){
        p=0;
        pos=0;
        k=0;
        system ("clear");
        printw ("Ingrese la contrasenia:\n");
        while(true){
            ch=getch();
            if (ch == intro) {
                break;
            }
            else{
                printw("%c", '*');
                password[pos++] = ch;
                password[pos] = '\0';
                k++;
            }
        }
        for(i=0;i<tam;i++) {
            if(k>=6){
                printw("Contrasenia no valida \n");
                p=1;
                break;
            }
            if(password[i]!=contra[i]) {
                printw ("Contrasenia no valida \n");
                p=1;
            }
        }
    }
}
```

```

        break;
    }
}
if(p==0) {
    printw ("\nBIENVENIDO AL SISTEMA\n");
    psw=1;
    break;
}

}

system("clear");
if(p==1) {

    printw("\nDemasiados Intentos");

}

getch();
echo();
endwin();
}

```

Configuracion de entradas y salidas con EasyPIO

Lo segundo que ejecutaremos en nuestro programa sera la configuracion de leds, pulsadores y llaves de nuestra Raspberry Pi por medio de la libreria EasyPIO. Este programa inicializa todo lo que usaremos en las secuencias de luces solicitadas.

```
#include "easypio.h"

extern int led[],pulsadores[],llaves[];

void Configuracion(void) {

    piolnit();

    //Inicializacion de leds
    for (int i=0;i<8;i++){
        pinMode (led[i],OUTPUT);
    }

    //Inicializacion de pulsadores
    for (int i=0;i<2;i++){
        pinMode (pulsadores[i],INPUT);
    }

    //Inicializacion de llaves
    for (int i=0;i<4;i++){
        pinMode (llaves[i],INPUT);
    }
}
```

Contador

Antes de ir a los codigos de las secuencias de luces explicaremos el programa Contador que usaremos en todas estas secuencias ya que sera el control para cada una de ellas, ya sea para finalizar los programas o para aumentar y disminuir su velocidad. Este constara de dos tipos de manejo, puede ser manejado de forma local desde Raspberry Pi o tambien manejado de forma remota desde una PC por medio de comunicaci3n en serie (Esta sera explicada mas adelante con sus respectivos codigos).

```
#include "easypio.h" Conclusi3n
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <wiringSerial.h>
#include <errno.h>
#include <string.h>

extern int led[],pulsadores[],llaves[],m,contador,serial_port;

int Contador(void) {
    int i=0,j=0,b=0;
    char dat='w'; //Inicializamos con un valor que no seleccione ningun opcion
    for (int c=0;c<contador;c++) {
        //Manejo desde pc
        if (serialDataAvail(serial_port)) //Verifica que el puerto este disponible devolviendo el numero 1, en caso contrario -1
        {
            dat = serialGetchar (serial_port); //Retorna el proximo caracter disponible en el puerto
        }
        if (dat=='s' || dat=='u' || dat=='d') //s:salir u:up d:down
        {
            if(dat=='s')
            { for(int ba=0;ba<8;ba++)
              {
                digitalWrite (led[ba],0);
              }
              return 1; //Si ponemos " exit (1) " sale de todo el programa, y estamos buscando que salga de la secuencia.
            }
            if (dat=='d' && contador > 10000) //Disminuye el delay producido por contador
            {
                contador= contador - 5000;
                printf("%d\n",contador);
                i++;
            }
            if (dat=='u' && contador < 90000) //Aumenta el delay producido por contador
```

```

        {
            contador= contador + 5000;
            printf("%d\n",contador);
            j++;
        }
    }

    //Manejo desde Raspberry

    if ((digitalRead(pulsadores[0]) && digitalRead(pulsadores[1])) ) {
        for(int ba=0;ba<8;ba++) {
            digitalWrite (led[ba],0);
        }
        return 1; //Volver al menu principal
    }

    if (digitalRead(pulsadores[0]) && i<1 && contador < 90000 ) //Aumenta el delay producido por contador
    {
        contador= contador + 5000;
        printf("%d\n",contador);
        i++;
    }

    if (digitalRead(pulsadores[1]) && j<1 && contador>10000)//Disminuye el delay producido por contador
    {
        contador= contador - 5000;
        printf("%d\n",contador);
        j++;
    }

}

return 0;
}

```

Secuencias de luces

Autofantastico

```
#include "easypio.h"
#include <stdio.h>
#include <stdlib.h>

extern int led[],pulsadores[],llaves[],m;
int Contador(void);

void AutoFantastico (void) {
    int i;
    system("clear");
    printf ("Auto Fantastico\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    while (m==0) //condicion para que siga ejecutandose la secuencia
    {
        for (i=0;i<8;i++) //recorrido de led 0 al 7
        {
            digitalWrite (led [i], 1);
            m=Contador(); //mientras la funcion contador no devuelva un 1 seguira ejecutandose
            if(m==1) {
                break;
            }
            digitalWrite (led [i], 0);
            m=Contador();
            if(m==1) {
                break;
            }
        }
        for (i=7;i>=0;i--) //recorrido de led 7 al 0
        {
            digitalWrite (led [i], 1);
            m=Contador();
            if(m==1) {
                break;
            }
            digitalWrite (led [i], 0);
            m=Contador();
            if(m==1) {
                break;
            }
        }
    }
}
```

```

    }

m=0;
system("clear");
}

```

Carrera

```

#include "easypio.h"
#include <stdio.h>
#include <stdlib.h>

extern int led[],pulsadores[],llaves[],m;
int Contador(void);

void Carrera(void) {
    int h,k;
    int tabla [16][8]= {
        {1,0,0,0,0,0,0,0},
        {1,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,0},
        {0,0,1,0,0,0,0,0},
        {0,0,1,0,0,0,0,0},
        {0,0,0,1,0,0,0,0},
        {0,0,0,1,0,0,0,0},
        {1,0,0,0,1,0,0,0},
        {0,1,0,0,1,0,0,0},
        {0,0,1,0,0,1,0,0},
        {0,0,0,1,0,1,0,0},
        {0,0,0,0,1,0,1,0},
        {0,0,0,0,0,1,1,0},
        {0,0,0,0,0,0,1,1},
        {0,0,0,0,0,0,0,1}
    };

    system ("clear");
    printf ("Carrera\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    while (m==0) {
        for (k=0;k<16;k++) //Recorrido de filas
        {
            m=Contador();
            if(m==1){
                break;
            }
        }
        for (h=0;h<8;h++) //Recorrido de columnas
        {

```



```

        digitalWrite (led [h],tabla [k][h]);
    }
}

}

m=0;
system("clear");
}

```

Apilada

```

#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"

int Contador(void);
extern int led[],m;

void Apilada(void){
    int i,j;
    system ("clear");
    printf ("Apilada\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    while(m==0){
        for(j=7;j>-1;j--){
            for(i=0;i<=j;i++){
                digitalWrite(led[i],1);
                m=Contador();
                if(m==1){
                    break;
                }
                digitalWrite(led[i],0);
            }
            if(m==1){
                break;
            }
            digitalWrite(led[j],1);
            delayMillis(200);
            digitalWrite(led[j],0);
            delayMillis(200);
            digitalWrite(led[j],1);
            delayMillis(200);
            digitalWrite(led[j],0);
            delayMillis(200);
            digitalWrite(led[j],1);
        }
        for(i=0;i<8;i++){
            digitalWrite(led[i],0);
        }
    }
    m=0;
    system ("clear");
}

```

Choque

```
#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"

extern int led[],pulsadores[],llaves[],m;
int Contador(void);

void Choque(void){
    int i=0;
    system ("clear");
    printf ("El Choque\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    while(m==0) {
        for(i=0;i<4;i++) //recorrido hacia adentro
        {
            digitalWrite(led[i],1);
            digitalWrite(led[7-i],1);
            m=Contador();
            if(m==1){
                break;
            }
            digitalWrite(led[i],0);
            digitalWrite(led[7-i],0);
            m=Contador();
            if(m==1){
                break;
            }
        }
        for(i=3;i>=0;i--) //recorrido hacia afuera
        {
            digitalWrite(led[i],1);
            digitalWrite(led[7-i],1);
            m=Contador();
            if(m==1){
                break;
            }
            digitalWrite(led[i],0);
            digitalWrite(led[7-i],0);
            m=Contador();
            if(m==1){
                break;
            }
        }
    }
    m=0;
    system ("clear");
}
```

Cortina

```
#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"

extern int led[],pulsadores[],llaves[],m;
int Contador(void);
void Cortina(void) {
    int i;
    system ("clear");
    printf ("Cortina\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    while(m==0) {
        for(i=0;i<8;i++) //Enciendo todos los leds
        {
            digitalWrite(led[i],1);
            m=Contador();
            if(m==1){
                break;
            }
        }
        for(i=0;i<8;i++) //Apaga todos los leds
        {
            digitalWrite(led[i],0);
            m=Contador();
            if(m==1){
                break;
            }
        }
        m=0;
        system ("clear");
    }
}
```

Gusano

```
#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"

extern int led[],pulsadores[],llaves[],m;
int Contador(void);
void Gusano(void){
    int i,k;
    system ("clear");
```

```

printf ("Gusano\n");
printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
int tabla[15][8]= {
    {0,0,0,0,0,0,0,0},
    {1,0,0,0,0,0,0,0},
    {0,1,0,0,0,0,0,0},
    {1,0,1,0,0,0,0,0},
    {0,1,0,1,0,0,0,0},
    {1,0,1,0,1,0,0,0},
    {0,1,0,1,0,1,0,0},
    {1,0,1,0,1,0,1,0},
    {0,1,0,1,0,1,0,1},
    {0,0,1,0,1,0,1,0},
    {0,0,0,1,0,1,0,1},
    {0,0,0,0,1,0,1,0},
    {0,0,0,0,0,1,0,1},
    {0,0,0,0,0,0,1,0},
    {0,0,0,0,0,0,0,1},
};
while(m==0) {
    for(k=0;k<15;k++) //Recorrido de filas
    {
        for(i=0;i<8;i++) //Recorrido de columnas
        {
            digitalWrite(led[i],tabla[k][i]);
            m=Contador();
            if(m==1){
                break;
            }
        }
        if(m==1){
            break;
        }
    }
    m=0;
    system("clear");
}

```

Barrido

```

#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"

extern int led[],pulsadores[],llaves[],contador,m;
int Contador(void);

```

```

void Barrido(void){

    int i,k;
    system ("clear");
    printf ("Barrido\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    int tabla[17][8]= {
        {0,0,0,0,0,0,0,0},
        {1,0,0,0,0,0,0,0},
        {1,1,0,0,0,0,0,0},
        {0,1,1,0,0,0,0,0},
        {0,0,1,1,0,0,0,0},
        {1,0,0,1,1,0,0,0},
        {1,1,0,0,1,1,0,0},
        {0,1,1,0,0,1,1,0},
        {0,0,1,1,0,0,1,1},
        {1,0,0,1,1,0,0,1},
        {1,1,0,0,1,1,0,0},
        {0,1,1,0,0,1,1,0},
        {0,0,1,1,0,0,1,1},
        {0,0,0,1,1,0,0,1},
        {0,0,0,0,0,1,1,0},
        {0,0,0,0,0,0,1,1},
        {0,0,0,0,0,0,0,1},
    };
    while(m==0) {
        for(k=0;k<17;k++) //Recorrido de filas
        {
            if(m==1){
                break;
            }
            for(i=0;i<8;i++) //Recorrido de columnas
            {
                digitalWrite(led[i],tabla[k][i]);
            }
            m=Contador();
            if(m==1){
                break;
            }
        }
    }
    m=0;
    system ("clear");
}

```

Navidad

```
#include <stdio.h>
#include <stdlib.h>
#include "easypio.h"

extern int led[],pulsadores[],llaves[],m;
int Contador(void);
void Navidad(void){
    int i;
    system ("clear");
    printf ("Navidad\n");
    printf ("Manejo local:\nPara salir presione los dos pulsadores\nManejo Remoto:\nPara salir presione s\n");
    while(m==0) {
        for(i=0;i<8;i=i+2) //Encendido de leds pares y apagado de led impares
        {
            digitalWrite(led [i],1);
            digitalWrite(led [i+1],0);
        }
        m=Contador();
        if(m==1){
            break;
        }
        for(i=1;i<8;i=i+2) //Encendido de leds impares y apagado de led pares
        {
            digitalWrite(led [i-1],0);
            digitalWrite(led [i],1); }
        m=Contador();
        if(m==1){
            break;
        }
    }
    m=0;
    system ("clear");
}
```

Comunicación serie

Receptor de datos en Raspberry

Primero explayaremos como realizamos el código para recibir datos en Raspberry para controlar las distintas secuencias solicitadas.

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <wiringSerial.h>
#include <errno.h>
#include <string.h>

extern int serial_port;

void Comunicacion_serie(void) {

    int aux,p=0;
    char str;
    serial_port = serialOpen ("/dev/serial0", 9600); //Abre el puerto serie si todo esta en orden
    if ( serial_port < 0){
        fprintf (stderr, "ERROR: %s\n", strerror (errno));
    }
    if (wiringPiSetup () == -1){ //Inicia configuracion de wiringPi
        fprintf (stdout, "ERROR: %s\n", strerror (errno)) ;
    }
    do{
        system ("clear");
        printf ("Comunicacion serie:\n\nIngresa un numero \n\t1-Auto Fantastico\n\t2-Carrera\n\t3-Apilada\n\t4-Choque\n\t5-Cortina\n\t6-Gusano\n\t7-Barrido\n\t8-Navidad\n\n\t9-Salir\n");
        printf ("Recibiendo datos desde el otro dispositivo\n");
        str= serialGetchar(serial_port); //Almacena el dato en la variable str
        printf ("%c", str); serialFlush(serial_port); //Limpia el bus de datos
        switch(str){
            case '1': {
                aux=contador; //Variable auxiliar para respaldar la variable original de contador
                AutoFantastico();
                contador=aux; //Respaldo
                break;
            }
            case '2': {
```

```

        aux=contador;
        Carrera();
        contador=aux;
        break;
    }
    case '3': {
        aux=contador;
        Apilada();
        contador=aux;
        break;
    }
    case '4': {
        aux=contador;
        Choque();
        contador=aux;
        break;
    }
    case '5': {
        aux=contador;
        Cortina();
        contador=aux;
        break;
    }
    case '6': {
        aux=contador;
        Gusano();
        contador=aux;
        break;
    }
    case '7': {
        aux=contador;
        Barrido();
        contador=aux;
        break;
    }
    case '8': {
        aux=contador;
        Navidad();
        contador=aux;
        break;
    }
    case '9': {
        p=1; //Asigna el valor 1 a p para salir de la sentencia do/while
        break;
    }
    default: {
        printf ("\nSeleccion no valida\n\n");
        delay(1000);
        break;
    }
}
}while(p!=1);
}

```


Transmisor de datos desde PC

El siguiente código es utilizado para enviar datos tanto al receptor de datos de raspberry como al programa Contador para subir velocidad, bajar velocidad o salir de la secuencia que se está ejecutando.

```
#include < curses.h>
#include < termios.h>

int main(){
    int fd,a=0,ch,q;
    char c[2];
    char opt;

    //CREAR ARCHIVO
    //O_WRONLY abre el archivo "/dev/ttyUSB0" en modo escritura
    //S_IRUSR|S_IWUSR son los modos que seleccione: Lectura y Escritura para el usuario
    fd = open("/dev/ttyUSB0",O_WRONLY|S_IRUSR|S_IWUSR);

    if (fd != -1){ //Verificamos si hay error al abrir el puerto
        do{
            system("clear");
            printf ("Comunicacion serie:\n\nIngrese un numero \n\t1-Auto Fantastico\n\t2-Carrera\n\t3-Apilada\n\t4-Choque\n\t5-Cortina\n\t6-Gusano\n\t7-Barrido\n\t8-Navidad\n\n\t9-Salir\n");
            ch='x';
            a=0;
            scanf ("%s",&opt);
            switch(opt){
                case '1': {
                    strcpy(c,"1");
                    write(fd,&c,sizeof(c));
                    fflush(stdin);
                    system("clear");
                    initscr(); //Iniciamos el modo Ncurses
                    clear();
                    printw ("La secuencia se esta ejecutando\n");
                    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tS=Salir\n");
                    echo();
                    do{
                        q=getch(); //ESC o s
                        if(q==115) // ASCII de s
                        {
                            strcpy(c,"s");
                            write(fd,&c,sizeof(c));
                            fflush(stdin);
                            a=1;
                        }
                    }
                    else{
                        getch(); //I
                        ch=getch(); // A o B
                        if(ch==65) //ASCII de A
                        {
                            strcpy(c,"u"); //Copia s a la variable c
                            write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
                        }
                    }
                }
            }
        } while (a==0);
    }
```

```

        fflush(stdin); //Limpia el bus de datos
    }
    if(ch==66) //ASCII de B
    {
        strcpy(c,"d");
        write(fd,&c,sizeof(c));
        fflush(stdin);
    }
}
}while(a!=1);
endwin();
break;
}
case '2': {
    strcpy(c,"2");
    write(fd,&c,sizeof(c));
    fflush(stdin);
    system("clear");
    initscr(); //Iniciamos el modo Ncurses
    clear();
    printw ("La secuencia se esta ejecutando\n");
    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tts=Salir\n");
    echo();
    do{
        q=getch(); //ESC o s
        if(q==115) // ASCII de s
        {
            strcpy(c,"s");
            write(fd,&c,sizeof(c));
            fflush(stdin);
            a=1;
        }
        else{
            getch(); //[
            ch=getch(); // A o B
            if(ch==65) //ASCII de A
            {
                strcpy(c,"u"); //Copia s a la variable c
                write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
                fflush(stdin); //Limpia el bus de datos
            }
            if(ch==66) //ASCII de B
            {
                strcpy(c,"d");
                write(fd,&c,sizeof(c));
                fflush(stdin);
            }
        }
    }while(a!=1);
    endwin();
    break;
}
case '3': {
    strcpy(c,"3");
    write(fd,&c,sizeof(c));
    fflush(stdin);

```

```

system("clear");
initscr(); //Iniciamos el modo Ncurses
clear();
printw ("La secuencia se esta ejecutando\n");
printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tS=Salir\n");
echo();
do{
    q=getch(); //ESC o s
    if(q==115) // ASCII de s
    {
        strcpy(c,"s");
        write(fd,&c,sizeof(c));
        fflush(stdin);
        a=1;
    }
    else{
        getch(); //[
        ch=getch(); // A o B
        if(ch==65) //ASCII de A
        {
            strcpy(c,"u"); //Copia s a la variable c
            write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
            fflush(stdin); //Limpia el bus de datos
        }
        if(ch==66) //ASCII de B
        {
            strcpy(c,"d");
            write(fd,&c,sizeof(c));
            fflush(stdin);
        }
    }
}while(a!=1);
endwin();
break;
}
case '4': {
    strcpy(c,"4");
    write(fd,&c,sizeof(c));
    fflush(stdin);
    system("clear");
    initscr(); //Iniciamos el modo Ncurses
    clear();
    printw ("La secuencia se esta ejecutando\n");
    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tS=Salir\n");
    echo();
    do{
        q=getch(); //ESC o s
        if(q==115) // ASCII de s
        {

```

```

        strcpy(c,"s");
        write(fd,&c,sizeof(c));
        fflush(stdin);
        a=1;
    }
else{
    getch(); //[
    ch=getch(); // A o B
    if(ch==65) //ASCII de A
    {
        strcpy(c,"u"); //Copia s a la variable c
        write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
        fflush(stdin); //Limpia el bus de datos
    }
    if(ch==66) //ASCII de B
    {
        strcpy(c,"d");
        write(fd,&c,sizeof(c));
        fflush(stdin);
    }
    }
}while(a!=1);

endwin();

break;
}

case '5': {

    strcpy(c,"5");
    write(fd,&c,sizeof(c));
    fflush(stdin);
    system("clear");
    initscr(); //Iniciamos el modo Ncurses
    clear();
    printw ("La secuencia se esta ejecutando\n");
    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tS=Salir\n");
    echo();
    do{
        q=getch(); //ESC o s
        if(q==115) // ASCII de s
        {
            strcpy(c,"s");
            write(fd,&c,sizeof(c));
            fflush(stdin);
            a=1;
        }
        else{
            getch(); //[
            ch=getch(); // A o B
            if(ch==65) //ASCII de A

```

```

        {
            strcpy(c,"u"); //Copia s a la variable c
            write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
            fflush(stdin); //Limpia el bus de datos
        }
        if(ch==66) //ASCII de B
        {
            strcpy(c,"d");
            write(fd,&c,sizeof(c));
            fflush(stdin);
        }
    }
}while(a!=1);

endwin();
break;
}

case '6': {
    strcpy(c,"6");
    write(fd,&c,sizeof(c));
    fflush(stdin);
    system("clear");
    initscr(); //Iniciamos el modo Ncurses
    clear();
    printw ("La secuencia se esta ejecutando\n");
    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tts=Salir\n");
    echo();
    do{
        q=getch(); //ESC o s
        if(q==115) // ASCII de s
        {
            strcpy(c,"s");
            write(fd,&c,sizeof(c));
            fflush(stdin);
            a=1;
        }
        else{
            getch(); //[
            ch=getch(); // A o B
            if(ch==65) //ASCII de A
            {
                strcpy(c,"u"); //Copia s a la variable c
                write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
                fflush(stdin); //Limpia el bus de datos
            }
            if(ch==66) //ASCII de B
            {
                strcpy(c,"d");
                write(fd,&c,sizeof(c));
            }
        }
    }while(a!=1);
}

```

```

        fflush(stdin);
    }
}
}while(a!=1);
endwin();
break;
}
case '7': {
    strcpy(c,"7");
    write(fd,&c,sizeof(c));
    fflush(stdin);
    system("clear");
    initscr(); //Iniciamos el modo Ncurses
    clear();
    printw ("La secuencia se esta ejecutando\n");
    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tts=Salir\n");
    echo();
    do{
        q=getch(); //ESC o s
        if(q==115) // ASCII de s
        {
            strcpy(c,"s");
            write(fd,&c,sizeof(c));
            fflush(stdin);
            a=1;
        }
        else{
            getch(); //[
            ch=getch(); // A o B
            if(ch==65) //ASCII de A
            {
                strcpy(c,"u"); //Copia s a la variable c
                write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
                fflush(stdin); //Limpia el bus de datos
            }
            if(ch==66) //ASCII de B
            {
                strcpy(c,"d");
                write(fd,&c,sizeof(c));
                fflush(stdin);
            }
        }
    }while(a!=1);
    endwin();
    break;
}

```

```

case '8': {

    strcpy(c,"8");

    write(fd,&c,sizeof(c));

    fflush(stdin);

    system("clear");

    initscr(); //Iniciamos el modo Ncurses

    clear();

    printw ("La secuencia se esta ejecutando\n");

    printw ("Opciones:\n\n\tUP=Subir Velocidad\n\tDOWN=Bajar Velocidad\n\tts=Salir\n");

    echo();

    do{

        q=getch(); //ESC o s
        if(q==115) // ASCII de s
        {
            strcpy(c,"s");
            write(fd,&c,sizeof(c));
            fflush(stdin);
            a=1;
        }
        else{
            getch(); //[
            ch=getch(); // A o B
            if(ch==65) //ASCII de A
            {
                strcpy(c,"u"); //Copia s a la variable c
                write(fd,&c,sizeof(c)); //Copia el contenido de c en fd
                fflush(stdin); //Limpia el bus de datos
            }
            if(ch==66) //ASCII de B
            {
                strcpy(c,"d");
                write(fd,&c,sizeof(c));
                fflush(stdin);
            }
        }
    }while(a!=1);

    endwin();

    break;

}

case '9': {

    strcpy(c,"9");

    write(fd,&c,sizeof(c));

    fflush(stdin);

    opt='0';

    break;

```

```
        }  
    }  
}while(opt!='0');  
printf("\nSalida");  
close(fd); //CERRAR ARCHIVO  
}
```


Velocidad inicial

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <wiringPiI2C.h>
#include <pcf8591.h>
#include "easypio.h"

#define Address 0x48
#define BASE 64
#define A BASE + 3

extern int pulsadores[], contador;

void Velocidad_inicial(void){

    int val;
    if (wiringPiSetup () == -1) {

        exit (1);
    }

    pcf8591Setup(BASE, Address);
    do{

        system("clear");
        printf ("Posicione el potenciómetro para seleccionar el contador inicial, luego pulse los dos pulsadores para salir\n");

        printf ("El contador tiene que ser mayor a 10000\n");
        val = (analogRead(A)*314)+10000; //Lectura de conversor y escalado
        printf ("Valor = %d \n", val);
        if(val<10000){ //Limite del valor inferior

            printf ("Seleccione otra velocidad porque es menor que la permitida\n");
            printf ("Velocidad = 10000\n");
            contador=10000;
        }
    }
    else{

        contador=val;
        printf("Velocidad = %d \n", val);
    }
}
```

```
    }  
    delay(200);  
    }while(!digitalRead(pulsadores[1])&&!digitalRead(pulsadores[0]));  
system("clear");  
printf ("La velocidad quedo fijada en: %d \n", contador);  
delay (1000);  
}
```

Conclusión

Para finalizar recalcamos el proyecto sobretudo por la introducción a programación sobre Raspberry y sus poderosas herramientas. Al empezar el proyecto aparte de volcar los contenidos desarrollados en la catedra debimos refrescarnos de conocimientos de materias anteriores como Informatica I y II que necesitamos para manipular Linux y las comunicaciones solicitadas. Tambien nos sirvio de buena practica de programación el hecho de realizar funciones por separado e incluirlas en un menu principal, ya que es muy comodo sobretudo para encontrar y solucionar errores.

Encontramos dificultades a la hora de conectar los programas externos al menu principal ya que al hacerlo de la forma tradicional nos daba errores, esto se soluciono incluyendolos con la función `#Include` a cada función en el menu principal. Utilizamos el metodo de testeo de cada función antes de incluirla en el menu principal para estar seguros que cada una funcione por separado.

El hecho de estar en una epoca especial de distanciamiento tambien dificulto la comunicación entre los integrantes y principalmente el intercambio y compra de materiales. Esto nos hizo demorar mas de lo debido ya que la solución a cada problema tenia doble esfuerzo.