**Vehicle Parking Slot Counter**

Aim: To implement vehicle parking slot counter system.

**Apparatus Used:** UET-01(LPC2148)

**Software Used:** Keil uVision (IDE to write code), Flash Magic (To upload the code in the processor)

**Project Description:**

Vehicle parking has always been a cumbersome in many sectors like offices, malls, etc. Many people find it difficult to search for the correct place to park their vehicles. Many of these sectors need to appoint a person to assist people where to park their vehicles.

So, our project basically focusses on reducing man effort of parking. Each parking slot should have a kind of a sensor which could indicate whether the slot is empty or full. There will be lights/LEDs which will show where the slot is empty or whether all the slots are completely filled.

We have implemented the counter system using LEDs and set a timer for each parking slot to be filled. After all the parking slots are filled, the LCD shows that the parking area is full.

We have introduced a system to show how many of the slots have been set free using matrix keys. Whenever a matrix key is pressed, it indicates that the certain place has been set free. So it will show that the the certain positions have been set free.

**Algorithm:**

The algorithm to solve the above problem is quite simple. We have used four components of the UET-01 kit which are as follows: LCD, LEDs, Buzzer, Relay. LEDs will show whether the slot is filled or not while LCD will display if the parking is full or not. Buzzer will be turned on at each slot filling while relay will switch after a row is filled so that it moves to the next row. Matrix(16 bit) is required to show that certain slot has been set free.

**Code:**

// KeyBoard Matrix

// In run mode keep SW1.1,2,3 bits in ON position. To conect the RS,RW,EN to the LCD module

// And for providing pull up for Rows and column of keypad matrix

// Keep SW2 in ON position to enable KeyPad H/W Module and

#include "LPC214x.h" /\* LPC21xx definitions \*/

#include <stdio.h>

#include <string.h>

#include "Disp\_Driver.H"

#include "KeyPad\_DD.h"

#include "GPIO.h"

#include "PinConnectBlock.h"

#include <lpc214x.h>

#include "C:\Users\notor\Desktop\finalprojectlast\lcd.h"

const unsigned int keyTable[] =

{

0x0001,//0b0000000000000001, //Key1

0x0002,//0b0000000000000010, //Key2

0x0004,//0b0000000000000100, //Key3

0x0008,//0b0000000000001000, //Key4

0x0010,//0b0000000000010000, //Key5

0x0020,//0b0000000000100000, //Key6

0x0040,//0b0000000001000000, //Key7

0x0080,//0b0000000010000000, //Key8

0x0100,//0b0000000100000000, //Key9

0x0200,//0b0000001000000000, //Key10

0x0400,//0b0000010000000000, //Key11

0x0800,//0b0000100000000000, //Key12

0x1000,//0b0001000000000000, //Key13

0x2000,//0b0010000000000000, //Key14

0x4000,//0b0100000000000000, //Key15

0x8000 //0b1000000000000000 //Key16

};

char str[17];

unsigned char MapKeyState(unsigned int kState);

void black(void);

void delay\_ms(unsigned int count);

int main(void)

{

char validKeyState;

unsigned int keyState,counter=0;

unsigned char currKey;

unsigned char prevKey;

PINSEL2 = 0x00000000;// Selected for GPIO

/\*Function: Initialize the LCD

Input: void

Output: void\*/

black();

Init\_LCD();

ClearDisplay();

InitKeyPad();

strcpy(str,"CAR PARKING SYS");

SetCharPosition(1,1);

WriteString((char\*)str);

strcpy(str,"FREE SLOT PRESS");

SetCharPosition(2,1);

WriteString((char\*)str);

Delay\_milliSecond(2000);

ClearDisplay();

InitKeyPad();

strcpy(str,"PRESS A KEY ...!");

SetCharPosition(1,1);

WriteString((char\*)str);

strcpy(str," ");

SetCharPosition(2,1);

WriteString((char\*)str);

Delay\_milliSecond(100);

currKey = 0;

prevKey = 0;

while(1) //Infinite loop

{

/\*

Function: This function must be called in the main loop

returns the current key state and key Press down

event.

LsB => Key1

....

MSB => Keyy16

Bit H => Currently Key is Pressed Down

And Key has Changed the state from released to pressed down

\*/

validKeyState = RunKeyPad();

if(validKeyState)

{

keyState = GetKeyState();

currKey = MapKeyState(keyState);

if(currKey != prevKey)

{

prevKey = currKey;

ClearDisplay();

strcpy(str,"THE SLOT NUMBER ");

SetCharPosition(1,1);

WriteString((char\*)str);

//foo1(&currKeyState);

if( currKey != 0)

{

counter++;

sprintf(str,(const char\*)" %-2u ARE FREE ",(unsigned int)counter);

}

else

{

strcpy(str,"IS => ");

}

SetCharPosition(2,1);

WriteString((char\*)str);

}

}

}

}

/\* Function : To Map the bit pattern to a key value as required

\* \*/

unsigned char MapKeyState(unsigned int kState)

{

unsigned char i;

unsigned char keyValue;

keyValue = 0;

for(i = 0; i <= 15; ++i)

{

if(kState == keyTable[i])

{

keyValue = i+1;

break;

}

}

return keyValue;

}

void delay\_ms(unsigned int count)

{

unsigned int j=0,i=0;

for(j=0;j<count;j++)

{

for(i=0;i<3000;i++);

}

}

void black(void)

{

int t=3;

unsigned int delay11;

while(t>=1){

IO0CLR = (1<<19);

for(delay11=0; delay11<800000; delay11++);

IO0SET = (1<<19);

for(delay11=0; delay11<800000; delay11++);

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<0);

delay\_ms(200);

IO0SET=(1<<0);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<1);

delay\_ms(200);

IO0SET=(1<<1);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<2);

delay\_ms(200);

IO0SET=(1<<2);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<4);

delay\_ms(200);

IO0SET=(1<<4);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<5);

delay\_ms(200);

IO0SET=(1<<5);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<6);

delay\_ms(200);

IO0SET=(1<<6);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<7);

delay\_ms(200);

IO0SET=(1<<7);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<8);

delay\_ms(200);

IO0SET=(1<<8);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<9);

delay\_ms(200);

IO0SET=(1<<9);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<10);

delay\_ms(200);

IO0SET=(1<<10);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

/\*IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

//IO0CLR = (1<<11);

delay\_ms(200);

IO0SET=(1<<11);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<12);

delay\_ms(200);

IO0SET=(1<<12);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<13);

delay\_ms(200);

IO0SET=(1<<13);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<14);

delay\_ms(200);

IO0SET=(1<<14);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<15);

delay\_ms(200);

IO0SET=(1<<15);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<16);

delay\_ms(200);

IO0SET=(1<<16);

delay\_ms(200);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;\*/

IO0CLR = (1<<19);

for(delay11=0; delay11<800000; delay11++);

IO0SET = (1<<19);

for(delay11=0; delay11<800000; delay11++);

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<0);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<1);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<2);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<4);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<5);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<6);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<7);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<8);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<9);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<10);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

/\*IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<11);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<12);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<13);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<14);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<15);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;

IODIR0 = 0xffffffff; //Configure the P1 pins as OUTPUT;

IO0CLR = (1<<16);

delay\_ms(400);

IOSET0 = 0x00f00000; // Make all the Port pins as high

delay\_ms(100);

IOCLR0=0x00f00000;\*/

t--;

}

strcpy(str,"CAR PARKING SYS");

SetCharPosition(1,1);

WriteString((char\*)str);

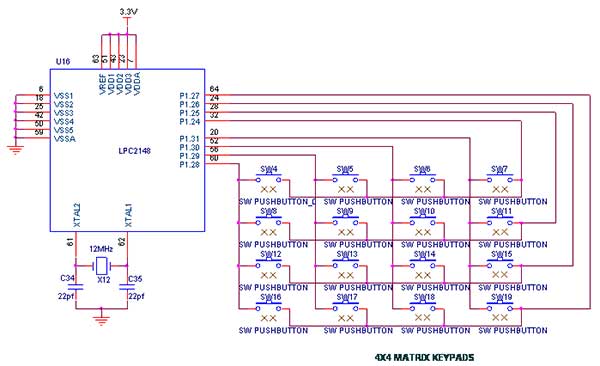
strcpy(str,"10 Slots Full");

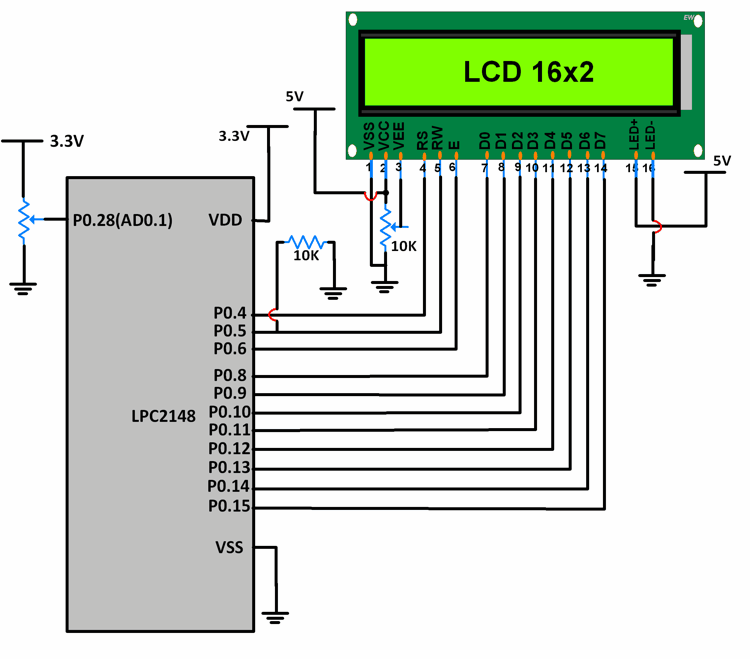
SetCharPosition(2,1);

WriteString((char\*)str);

}

**Pin Diagrams:**

****

****