(1)LED TOGGLING

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
3	LEDs and Resistors	1
5	Proteus Software and Keil Software	-

```
#include"lpc214x.h" // lpc2148 header file for KIEL IDE

void delay(unsigned long int h) // delay function

{

unsigned long int i,j;

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

for(i=0;i<5000;i++);

}

int main()

{

IODIR0 = 0xFFFFFFFF; // making port 0 output port

IODIR1 = 0xFFFFFFFF; // making port 1 output port

while(1)

{

IOCLR0 = 0xFFFFFFFF; // making port 0 low

IOCLR1 = 0xFFFFFFFF; // making port 1 low

delay(1000);

IOSET0 = 0xFFFFFFFF; // making port 0 low

IOSET1 = 0xFFFFFFFF; // making port 1 low

delay(1000);

}

}
```

(2)LED SWITCH

S.No	Hardware & Software Requirements	Quantiy
1	ARM Processor LPC2148	1
2	LEDs and Switch	1
3	Proteus Software and Keil Software	-

```
#include <lpc214x.h>

int i,b;

int main()

{

IODIR0=0x000000FF;

IODIR0=~(1<<16)&~(1<<23);

while(1)

{

if((IOPIN0&(1<<16))==0)

{

for(b=0;b<8;b++)

{

IOSET0=(1<<b);

for(i=0;i<120000;i++);

IOCLR0=(1<<b); for(i=0;i<120000;i++);

}

}

else if((IOPIN0&(1<<23))==0)

{

for(b=7;b>=0;b--)

{

IOSET0=(1<<b); for(i=0;i<120000;i++);

IOCLR0=(1<<b); for(i=0;i<120000;i++);

}

}

else

{

IOCLR0=0x000000FF;

}

}

}
```

(3)7 – SEGMENT

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
2	7seg LED	1
3	Proteus Software and Keil Software	-

```
#include<LPC21XX.H> //header file for lpc2148

void delay(); //delay function

unsigned int i, j, k; // global variable

unsigned int
ar[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0x
f8,0x80,0x90};

// array data

int main( )

{

PINSEL0=0X00000000; //select port0 as gpio mode

IO0DIR= 0X000000FF; // make starting 8 pin as
output and other as input of port 0

while(1)

{

for(i=0;i<10;i++)

{

IO0SET = ar [i];

delay ();

IO0CLR = ar [i];

}

}

return 0;

}

void delay ()

{

for(j=0;j<1000;j++)

for(k=0;k<400;k++);

}
```

(4)DC MOTOR

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
2	L293D IC	1
3	Proteus Software and Keil Software	-
4	Simple DC Motor active	1

```
#include<lpc214x.h>

#define bit(x) (1<<x)

#define delay for(i=0;i<=60000;i++)


unsigned int i;

int main()

{

  IO0DIR=0xf;              //Declaring as a output

  IO0PIN=0;                 //Clear all IO Pins in P0

  VPBDIV=0x01;              //PCLK = 60MHz

  while(1) {

    /*Forward*/

    IO0SET=bit(0);       //IN1 = 1

    IO0CLR=bit(1);       //IN2 = 0

    delay;delay;

    /*Off*/

    IO0CLR=bit(0)|bit(1); //IN1 = IN2 = 0

    delay;delay;

    /*Reverse*/

    IO0SET=bit(1);       //IN2 = 1

    IO0CLR=bit(0);       //IN0 = 1

    delay;delay;

    /*Off*/

    IO0CLR=bit(0)|bit(1); //IN1 = IN2 = 0

    delay;delay;

  }

}
```

(5)STEPPER MOTOR

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2138	1
2	ULN2003A	1
3	Proteus Software and Keil Software	-
4	Stepper Motor Animated Unipolar.	1

```
#include <lpc214x.h>

#define PULSE_DELAY 1000 // Adjust this value for
desired speed and step size

void delay(unsigned int count) {

  unsigned int i;

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

}

void stepperMotorControl() {

  unsigned int sequence[] = {0x01, 0x02, 0x04,
0x08}; // Define the sequence for 4 steps

 int i, j;

    while (1) {

     for (i = 0; i < 4; i++) {

         IO0PIN = sequence[i]; // Output the
sequence

         delay(PULSE_DELAY);   // Delay for motor
speed

     }

  }

}

int main() {

  PINSEL0 = 0x00000000; // Configure GPIO pins as
GPIO

 IODIR0 = 0x0000000F; // Set the lower 4 bits as
outputs

 stepperMotorControl(); // Start stepper motor
control

 return 0;

}
```

(6)LCD DISPLAY

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2138	1
2	LM016L	1
3	Proteus Software and Keil Software	-

```
#include<lpc214x.h>

#define bit(x) (1<<x)

#define delay for(i=0;i<1000;i++);

unsigned int i;

void lcd_int();

void dat(unsigned char);

void cmd(unsigned char);

void string(unsigned char *);

void main()

{

  IO0DIR|=0XFFF;

  lcd_int();

  cmd(0x8a);

  string(" HI! NAMASTE WELCOME ");

  while(1) {

    cmd(0x18);

    delay;

  }

}


void lcd_int()

{

  cmd(0x30);

  cmd(0x0c);

  cmd(0x06);

  cmd(0x01);
```

```
   cmd(0x80);

}


void cmd(unsigned char a)

{

   IO0PIN&=0x00;

   IO0PIN|=(a<<0);

   IO0CLR|=bit(8);             //rs=0

   IO0CLR|=bit(9);             //rw=0

   IO0SET|=bit(10);             //en=1

   delay;

   IO0CLR|=bit(10);             //en=0

}


void dat(unsigned char b)

{

   IO0PIN&=0x00;

   IO0PIN|=(b<<0);

   IO0SET|=bit(8);             //rs=1

   IO0CLR|=bit(9);             //rw=0

   IO0SET|=bit(10);             //en=1

   delay;

   IO0CLR|=bit(10);             //en=0

}


void string(unsigned char *p)

{

   while(*p!='\0') {

      dat(*p++);

   }

}
```

(7)UART

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2138	1
2	CRYSTAL	1
3	UART	2
4	Keil and Proteus Software	-

```
#include <LPC214x.H> // LPC21xx definitions
*/
char a;
void uart0_init(){

PINSEL0 = 0x00000005; // Enable RxD0 and
TxD0 */
U0LCR = 0x83; // 8 bits, no Parity, 1 Stop bit
*/
U0DLL = 97; // 9600 Baud Rate @ 15MHz
VPB Clock */
U0LCR = 0x03; // DLAB = 0 */
}
void uart0_putc(char c){
while(!(U0LSR & 0x20)); // Wait until UART0
ready to send character
U0THR = c; // Send character
}
int uart0_getc (void) {
while (!(U0LSR & 0x01));
return (U0RBR);
}
int main (void) {
uart0_init();
while (1) {
a=uart0_getc();
uart0_putc(a);
}
}
```

PROCEDURE :

1. Create a New project, Go to “Project” and close the current project “Close Project”.

2. Next Go to the Project New µvision Project Create New Project Select Device for Target.

3. Select the data base NXP LPC2148.

4. Add Startup file and Next go to “File” and click “New”.

5. Write a program on the editor window and save as “xxxxx.c”.

6. Add this source file to Group and click on “Build Target” or F7.

7. Create a Hex file from “Project” menu and click on “Rebuild all target Files”.

8. Create the circuit diagram in Proteus Software and upload the hex file of your program by double clicking on the microcontroller in Proteus workspace.

9. Debug and run your Proteus file to see the working of the circuit.