

LED BRIGHTNESS CONTROL LOGIC

Understanding the Problem statement:

For the control of the store led brightness. We are assuming the lights to be controlled by a knob (potentiometer) that controls the lights by varying at a particular values.

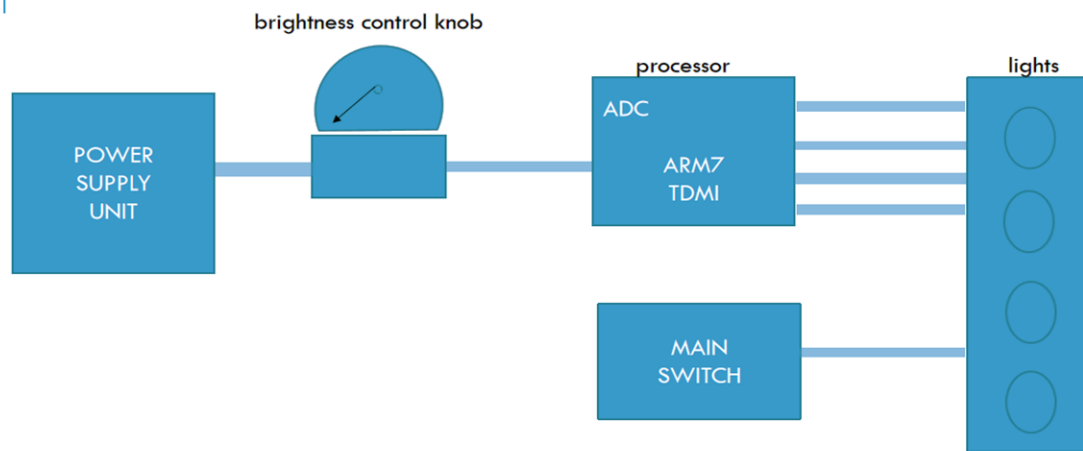
To control the led brightness Analog to Digital Converter is used to control the led on the development board.

The potentiometer values are accessed by the ADC data Register and through the value limits the led lights and to control the brightness is simply turning off the lights that aren't used.

Also a mains switch is used to use led's full brightness(turning all LED's ON and OFF) irrespective of the knob variations.

We are assuming brightness of led to be controlled by the Analog to digital converter which is already internally connected to the ARM LPC2148 development kit.

BLOCK DIAGRAM



To program the ADC0 having 6 channels, we are accessing ADC0.4

i.e., the fourth channel of Analog to digital converter [ADC0.4]
The already described reference voltage which is V_{ref} of 3.3V maximum for the potentiometer in the internal ADC which the variable voltage which is our analogous input to the ADC.

By controlling the potentiometer the ADC gets different voltage readings everytime when the potentiometer is varied accordingly.

The ADC converts the variable voltage into digital output. So the digital output which is generated by the analogous input voltage and the ADC which we are using with 10 bit resolution.

So the 10 bit data is accessed by AD0GDR.

From the data register the 10 bit data for different voltage values is nothing but our different brightness levels for the LED's .

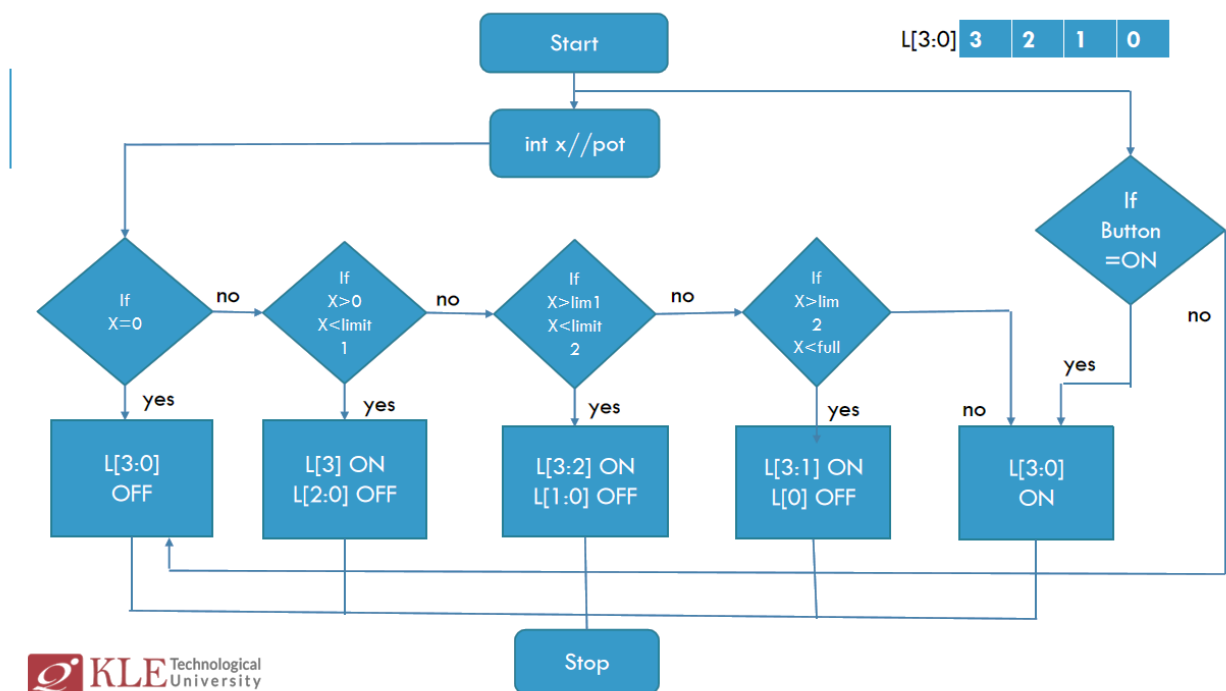
For different brightness levels like 25%, 50% and 75% there are different digital vlaues for these brightness levels LED's are controlled in such a way that for each brightness levels, simultaneously when the brightness keeps on increasing sequentially one LED turns ON.

So this implies that if the store is crowded the store's every LED should be turned ON,if the number of customers in store is less accordingly the LED's turn OFF based on the customers in the store.

If no customers are present in the store every LED's in the store should be completely turned OFF.

By having the main switch present in the store by default should be either turned ON or completely turned OFF.

FLOW CHART



CODE:

```
#include <lpc214x.h>

#include <stdio.h>

void delay();

int main (void)
{
    unsigned int result,temp;

    PINSEL1 = 0x00040000; //p0.25 pin is selected as as ad0.4

    IO0DIR = 0x000f0000;

    while(1)
    {
        AD0CR = 0x09200010; //control register for adc ad.04

        while((temp = AD0GDR) == 0x80000000);

        result = AD0GDR; //to get the adc values in result

        AD0CR &= ~0x01000000; //to stop the ADC

        result = (result >> 6);

        result = result & 0x000003FF;

        /*if(result == 0x00){

            IO0CLR = 0x00000000;

            IO0SET = 0x000F0000;

            delay();

        }*/
    }
}
```

```
if(result>0x0000 && result<=0x00AA){
```

```
IO0CLR = 0x00080000;
```

```
IO0SET = 0x00070000;
```

```
    delay();
```

```
    }
```

```
    else if(result>0x00AA && result<=0x0155){
```

```
        IO0CLR = 0x000C0000;
```

```
        IO0SET = 0x00030000;
```

```
        delay();
```

```
    }
```

```
    else if(result>0x0155 && result<=0x02AA){
```

```
        IO0CLR = 0x000E0000;
```

```
        IO0SET = 0x00010000;
```

```
        delay();
```

```
    }
```

```
    else{
```

```
        IO0CLR = 0x000F0000;
```

```
        IO0SET = 0x00000000;
```

```
        delay();
```

```
    }
```

```
}
```

```
}
```

```
void delay(){  
    unsigned int i;  
    for(i=0; i<100000; i++); }
```