Name: ..... Student ID: .....
Name: ..... Student ID: .....

Laboratory 04
Introduction to 7-Segment Display and Opto Sensor

1. Seven Segment Display

The ***7-segment display*** consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit to be displayed. An additional 8th LED allows the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individually LED pins are labeled from ***a*** through to ***g*** representing each individual LED. The other LED pins are connected together and wired to form a **common pin**.

The displays common pin is generally used to identify which type of 7-segment display it is. As each LED has two connecting pins, one called the "Anode" and the other called the "Cathode", there are therefore two types of LED 7-segment display called: **Common Cathode** (CC) and **Common Anode**(CA).

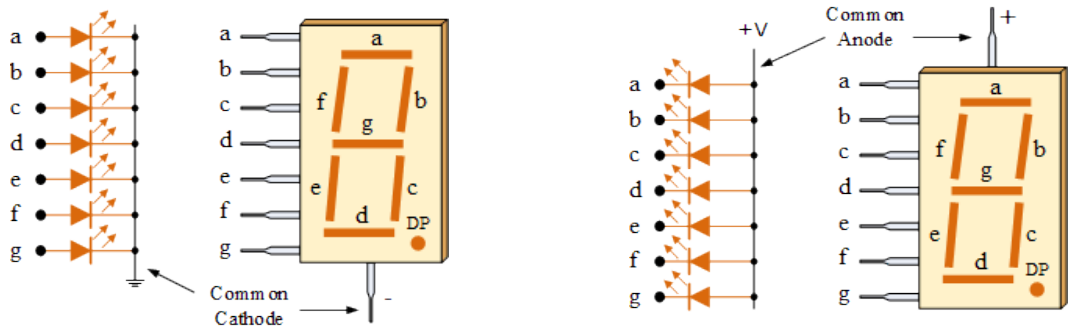


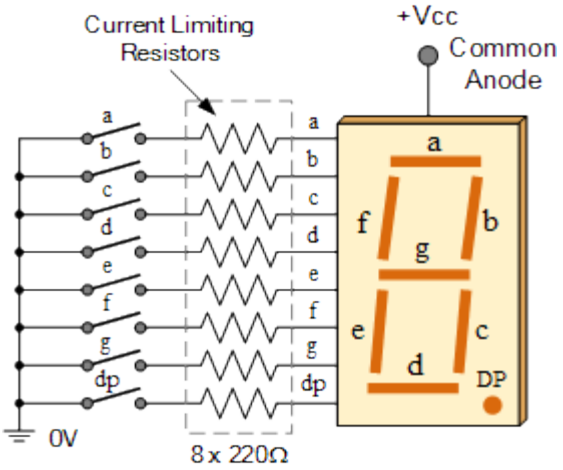
Fig.1 A common cathode 7-segment display and a common anode 7-segment display

1. The **Common Cathode** (CC) – All the cathode connections of the LED segments are joined together to logic "0" or ground. The individual segments are illuminated by application of a "HIGH", or logic "1" signal via a current limiting resistor to forward bias the individual Anode terminals (a-g).

2. The **Common Anode** (CA) – All the anode connections of the LED segments are joined together to logic "1". The individual segments are illuminated by applying a ground, logic "0" or "LOW" signal via a suitable current limiting resistor to the Cathode of the particular segment (a-g).

In general, common anode displays are more popular as many logic circuits can sink more current than they can source. Depending upon the decimal digit to be displayed, the particular set of LEDs is forward biased. For instance, to display the numerical digit 0, we will need to light up six of the LED segments corresponding to a, b, c, d, e and f. Then the various digits from 0 through 9 can be displayed using a 7-segment display as shown.

Driving a 7-segment Display



In this example, the segments of a common cathode display are illuminated using the switches. If switch "**a**" is closed, current will flow through the "**a**" segment of the LED to the current limiting resistor connected to pin **a** and to 0 volts, making the circuit. Then only **segment a** will be illuminated. So a LOW condition (switch to ground) is required to activate the LED segments on this common anode display.

Display value	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Segment drive (B)										
(MSB)	0011	0000	0101	0100	0110	0110	0111	0000	0111	0110
(LSB)	1111	0110	1011	1111	0110	1101	1101	0111	1111	1111
B (hex)	0x3F	0x06	0x5B	0x4F	0x66	0x6D	0x7D	0x07	0x7F	0x6F
Actual display										

Fig.2. The binary codes for control switches and its corresponding numbers

2. Opto Sensor

Opto sensor consists of a light emitting element which transforms an electrical signal to an optical signal, and a light receiving element which transforms the optical signal to an electrical signal. Both of these elements are housed in one package, making the opto sensor essentially a **non- contact switch**. Opto sensors detect presence, length, number, etc., of objects, using visible or infrared light which is blocked or reflected by the object.

2.1 Interrupter type opto sensors

This is the most popular type of opto sensors. Emitting and receiving elements are housed in one package and face each other. They sense presence of objects in the sensing slot, without physical contact.

2.2 Reflective type opto sensors.

Emitting and receiving elements are mounted on the same surface at a fixed angle to each other. They detect presence or position of an object through light reflection. Use of high contrast materials (e.g. white and black) is recommended to achieve high sensing resolution.

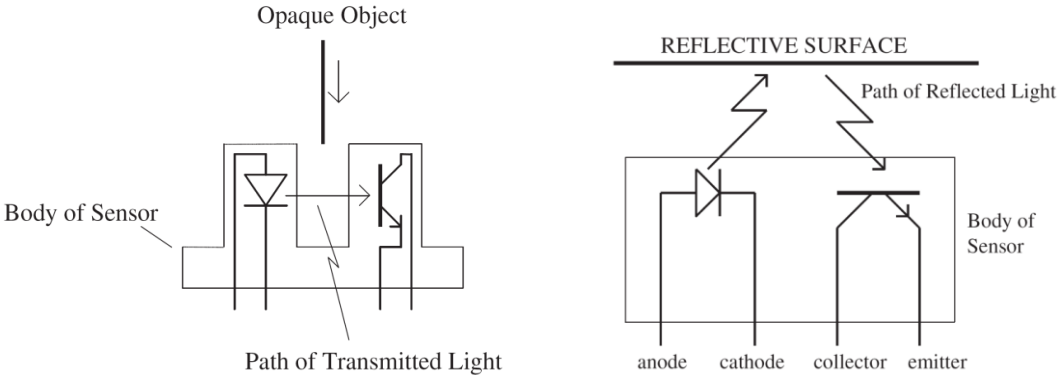


Fig.3. Transmissive opto-sensor and Reflective opto-sensor

2.3 Simple Drive Circuit for Opto Sensor

The simple drive circuit for opto-sensor is shown in Fig.4. In this circuit R1 limits amount of current flow to the diode. R2 is a pull-up resistor for a photo transistor. The circuit operates as follows. The transistor acts as a switch. When there is no opaque object between diode and transistor, it turns on and conducts current through R2. With an appropriate value of R2, V_C can be very closed to GND and gives logic LOW. When the transmitted light is blocked by an opaque object, the transistor is turned off. Once there is no current flow, V_C is equal to Vsupply and gives logic HIGH. Therefore we can use V_C to create a digital signal to MCU port.

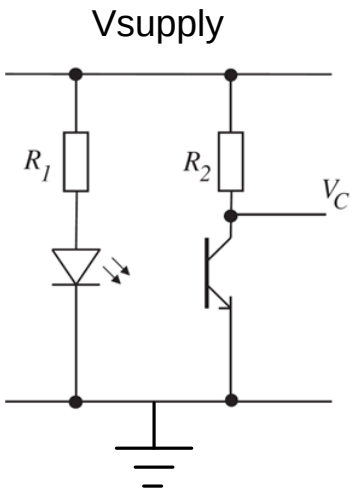


Fig.4. Simple drive circuit for opto-sensor

2.4 Connecting the devices to LPCXpresso board

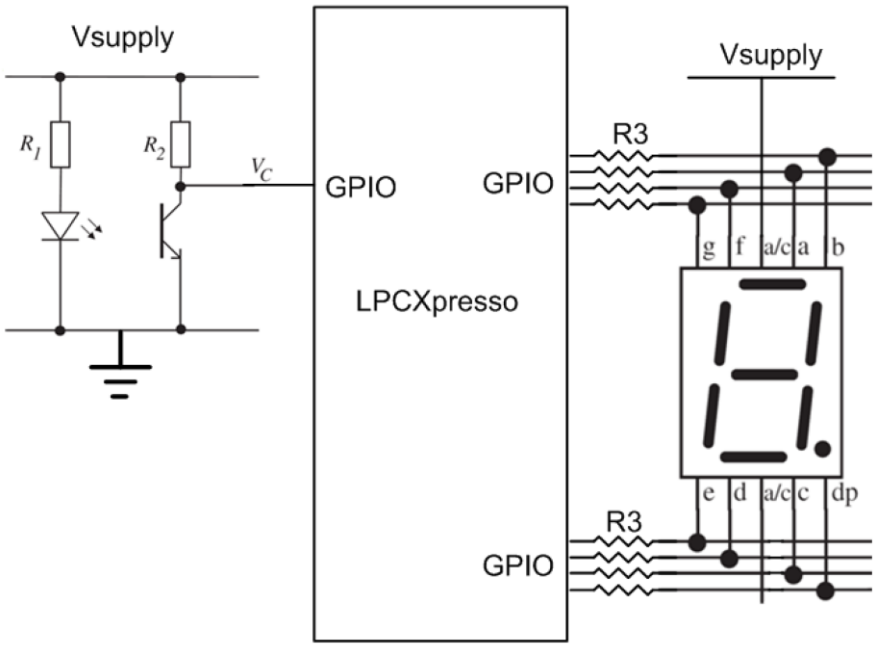


Fig.5 LPCXpresso connects to an opto sensor and
a common anode seven-segment using GPIO

In Fig.5 the seven-segment is a common anode type. Therefore the common pin (a/c) is connected to Vsupply. To light up the LED, GPIO needs to be in logic LOW. On the other hand, if the common cathode type is used, the common pin (a/c) must be connected to GND and GPIO needs to be in logic HIGH.

When need to display more than one digit, switches and digit selector signals are required. As shown in Fig. 6 and Fig.7 for the common anode and the common cathode respectively.

From Fig.6 each segment on each display is wired together, and connected back to a microcontroller pin configured as digital output. The common anode of each digit is then connected to its own switch using **BJT PNP Transistor A1015**. When a select digit signal is **LOW**, it turns on the switch to connect that digit to its Vsupply. Thus, it enables that digit to display.

The timing diagram shown then applies. The segment drives are configured for Digit 1, and that digit's drive transistor is activated, illuminating the digit. A moment later the segment drives are configured for Digit 2, and that digit's drive transistor is activated. This continues endlessly with each digit in turn. If it is done rapidly enough, then the human eye perceives all digits as being continuously illuminated; a useful rate is for each digit to be illuminated in turn for around 5 ms.

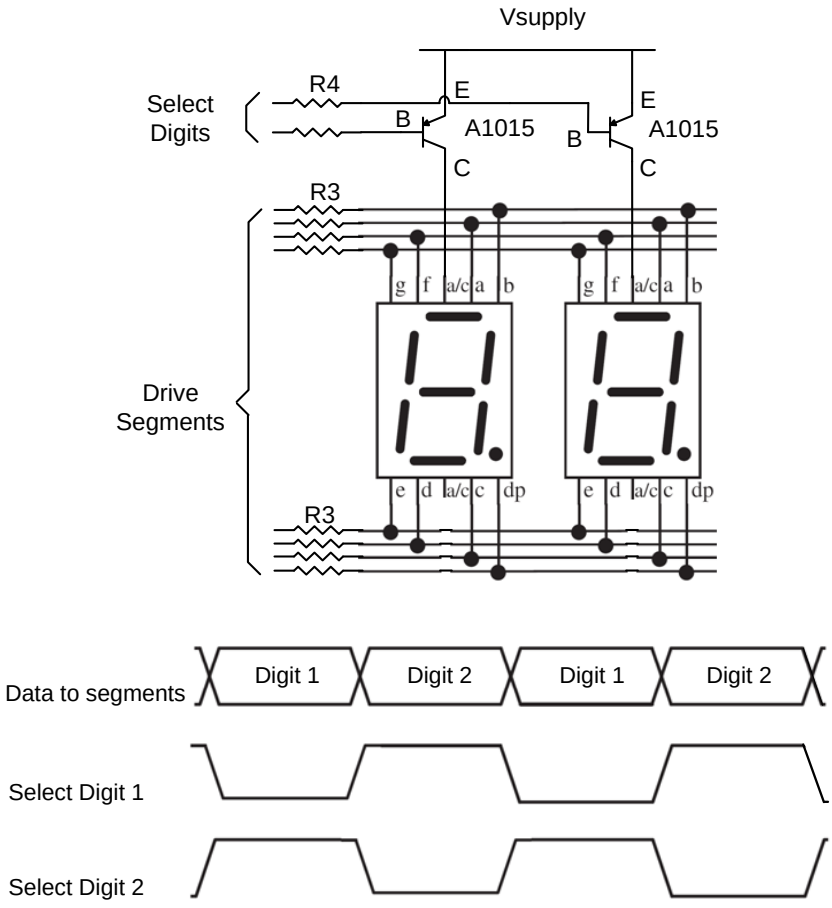


Fig. 6 Schematics and timing diagram for 2 digit display using the common anode

For the common cathode, the schematic and its timing diagram is shown in Fig.7. The differences are **BJT PNP Transistors C1815** are used instead and the logics to drive the segments and select the digits are opposite to those of the common anode.

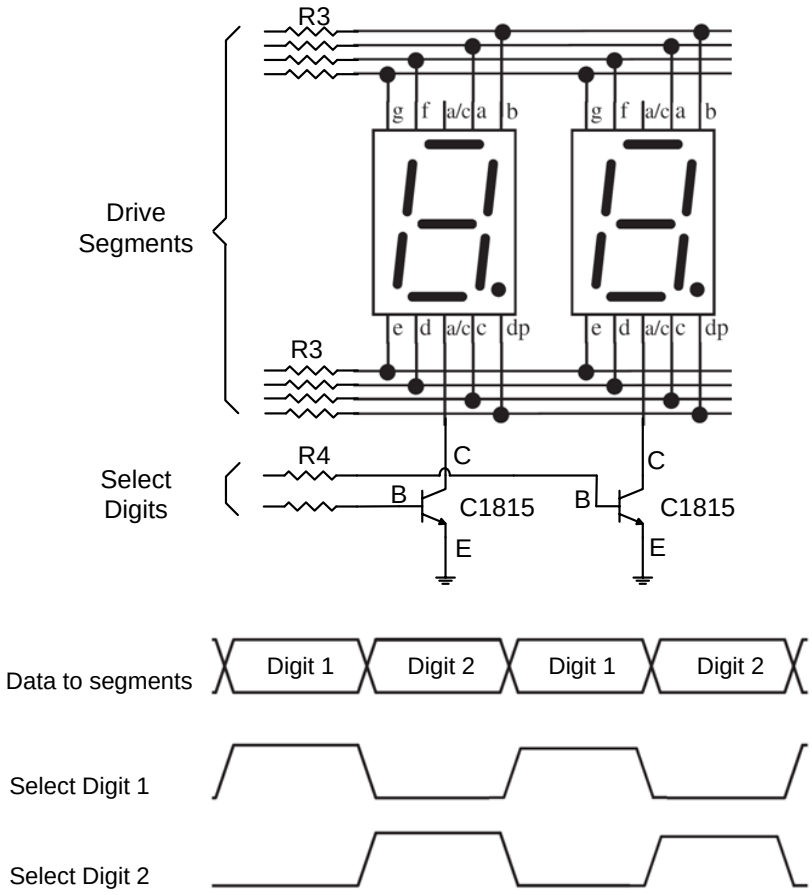


Fig. 7 Schematics and timing diagram for 2 digit display using the common cathode

3. The experiment

3.1 Create New Project

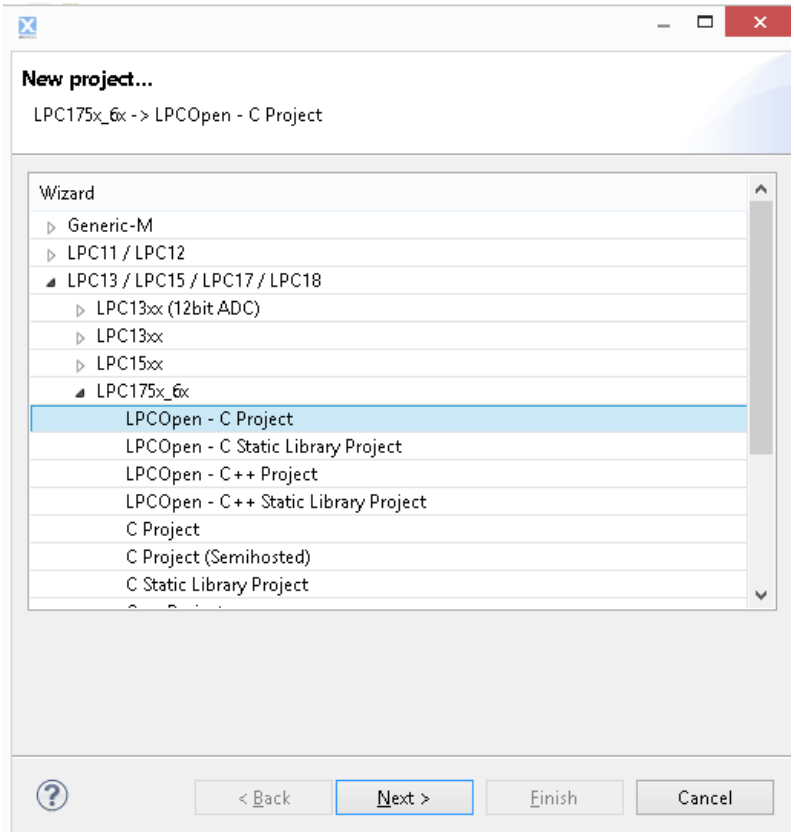
In quickstart panel (the lower left panel in your IDE's Develop view)

Select New Project

Wizard > LPC13 / LPC15 / LPC17 / LPC18

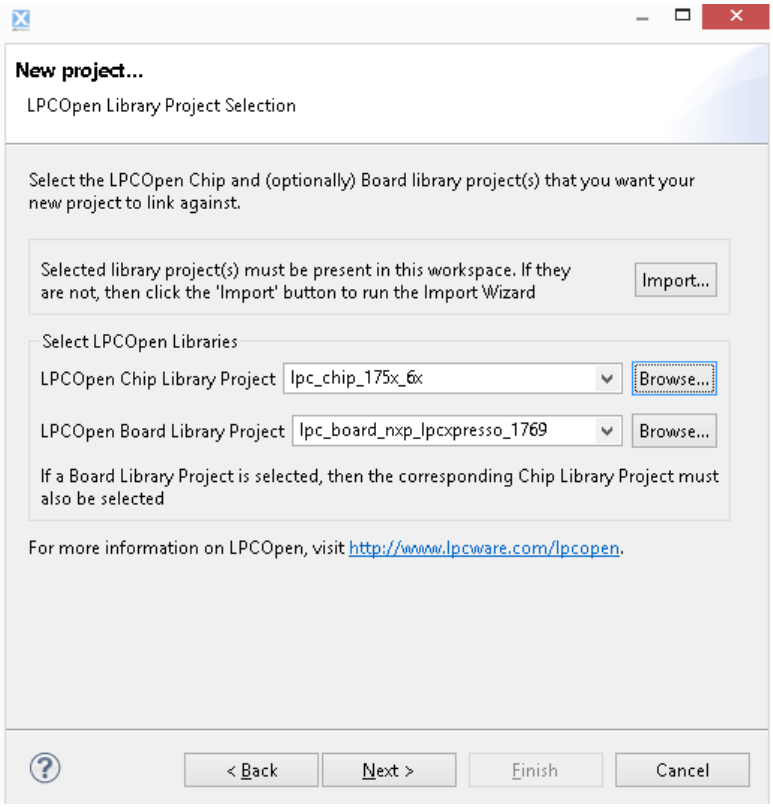
Select LPCOpen – C Project

Then click Next

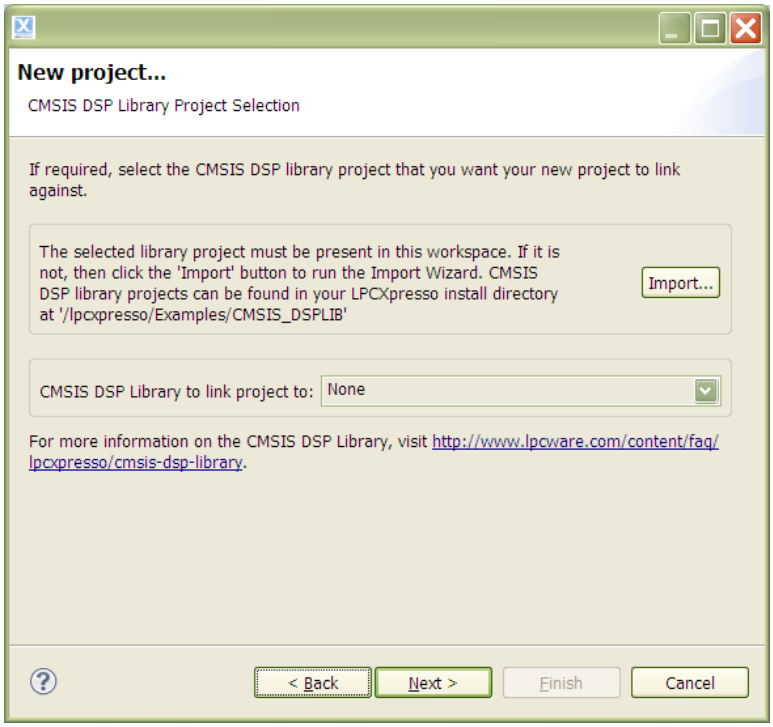


Fill in the **Project name**

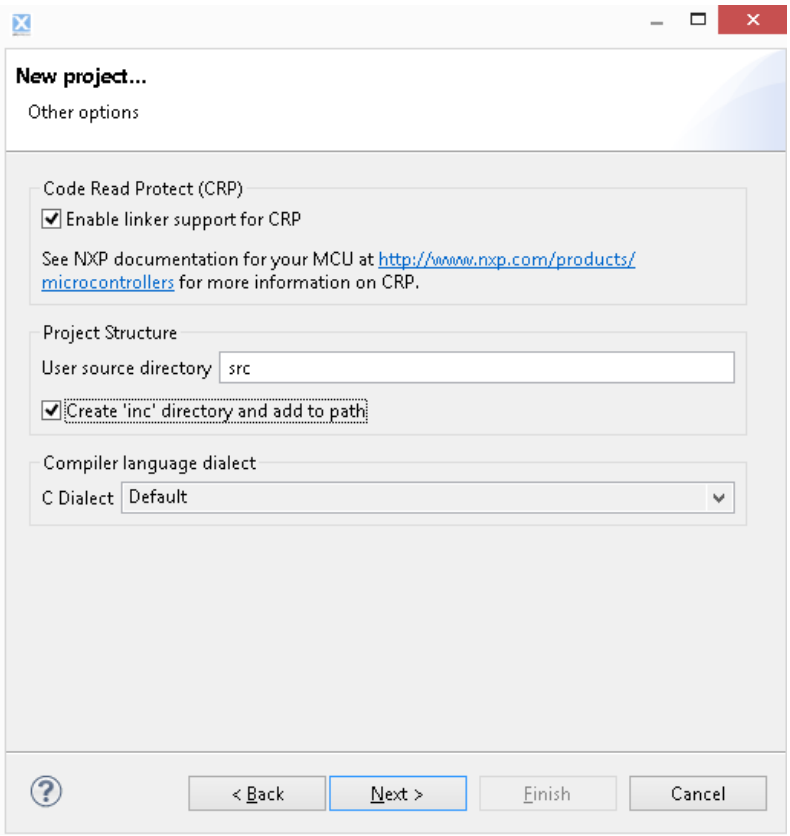
Select **LPC1769** as the target MCU. Then, click Next



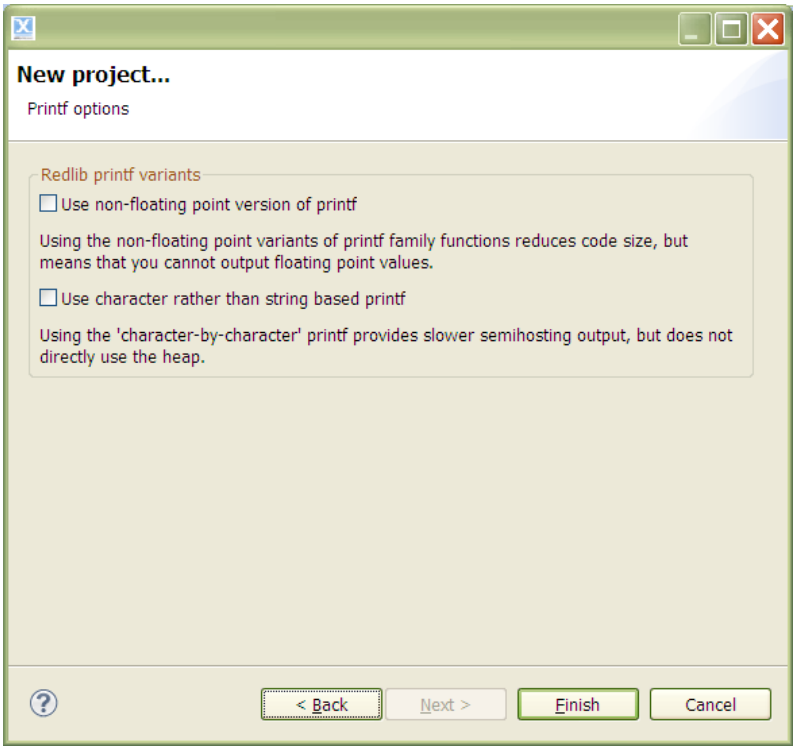
Select LPCOpen Chip Library Project to be **lpc_chip_175x_6x**
Select LPCOpen Board Library Project to be **lpc_board_nxp_lpcxpresso_1769**
Then, click Next



If you your project need CMSIS DSP library you can select library from the pull down menu. In this lab select **None.**
Then, click Next



Select Enable linker support for CRP and Create ‘inc’ directory and add to path.
Use Default C Dialect



After processing, your New Project will automatically appear in the **Project Explorer** (the top left panel of the Develop view)

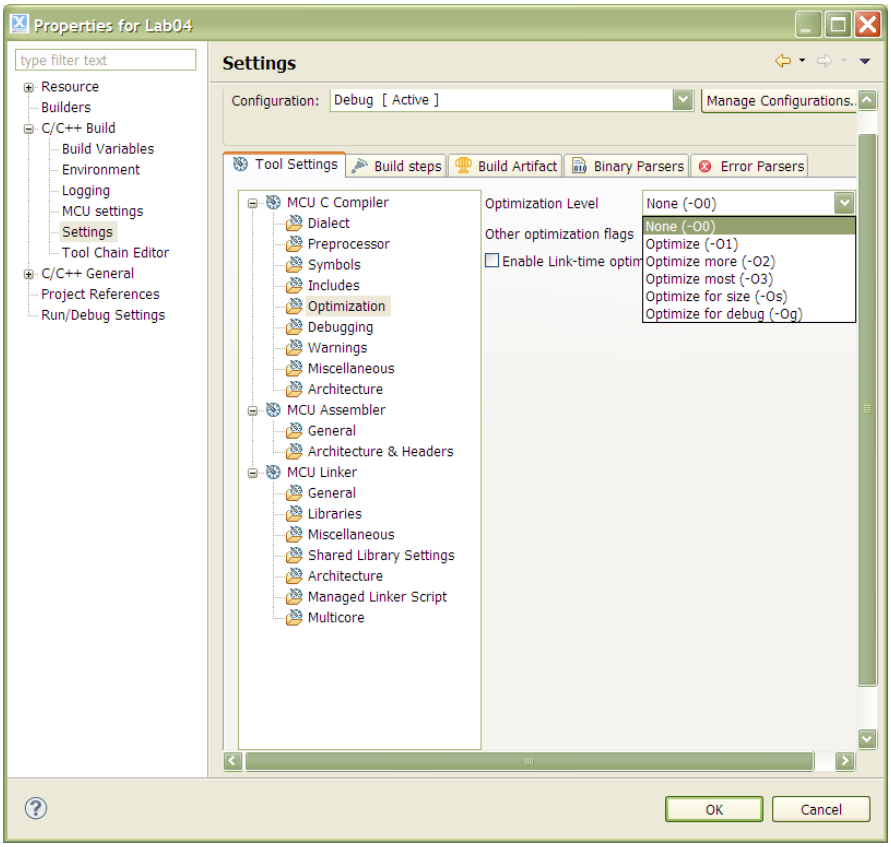
Setting Tool Optimization Level for the new project

Select your new project in **Project Explorer**

Right click > Properties -> C/C++ Build -> Setting -> Tool Setting -> MCU C Compiler -> Optimization

Change Optimization Level to **None**

(This because the default optimization level may not call some functions that are considered unnecessary)



Edit main program for your experiment. Note that by default the main program already includes the code that sets the on-board LED to the state of "on".

Example Program

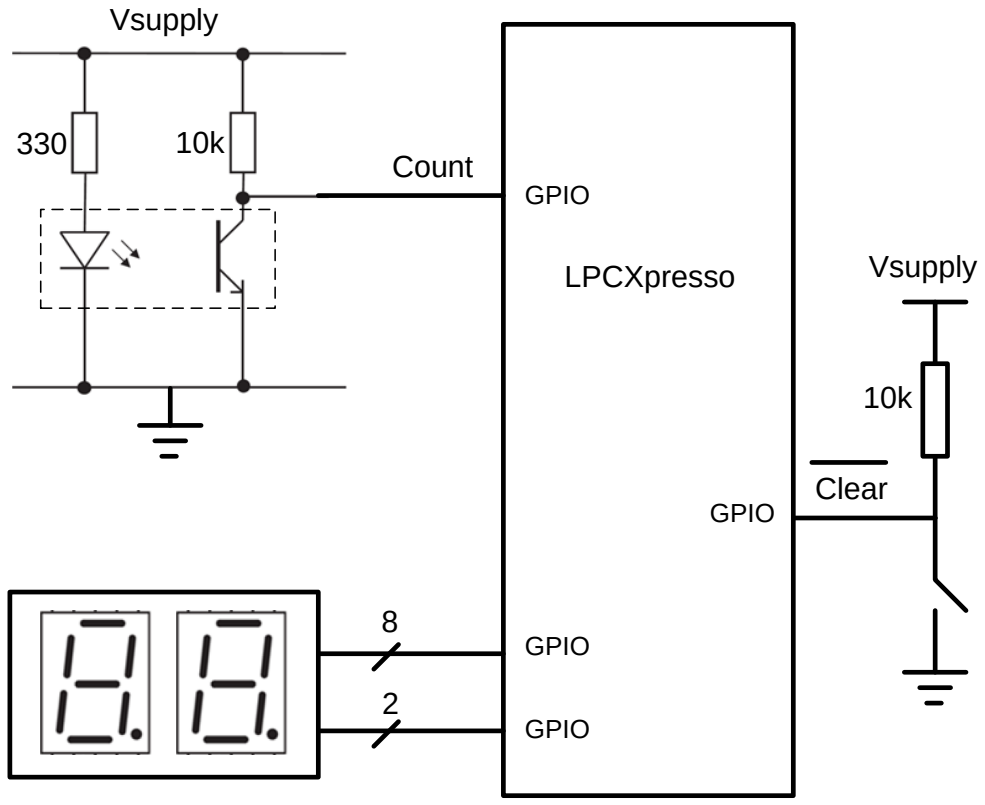
```
Void wait(void) {
      int i;
      for(i=0; i<10000000; i++);
}
int main() {
      Chip_GPIO_Init(0);       // Initialize GPIO Port 0
      Chip_GPIO_SetDir(LPC_GPIO, 0, 8, 1);// Port 0 Pin 8 for output

      Chip_GPIO_SetDir(LPC_GPIO, 0, 9, 1);
      Chip_GPIO_SetDir(LPC_GPIO, 0, 7, 0); // Port 0 pin 7 for input

      bool opto_Sensor = Chip_GPIO_GetPinState(LPC_GPIO, 0, 7);
      Chip_GPIO_WritePortBit(LPC_GPIO, 0, 8, true);

      wait();
}
```

3.2 Write a program to complete the following tasks

Base on Fig.8, every time an opaque object passing through the opto sensor, the program increases its counter by one and displays its current status at the seven-segment display. Whenever a user pushes a switch button, it clears the counter and the display flashes "0"

1.) Connect only one seven-segment display and count 0-9.

2.) Add another digit of seven-segment display and modify program to count from 00-99. To save the number of GPIO port used, you need to share pins between two digits by using selector signals and transistors as shown in Fig.6 and Fig.7 respectively.



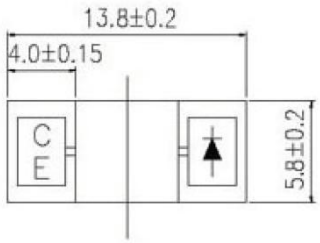
For seven-segment display use:
R3 = 880 or 1k; R4 = 330

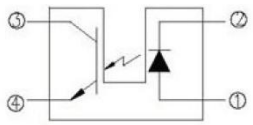
Connect the common pins (3 or 8) to **Vsupply** for common **anode**
Connect the common pins (3 or 8) to **GND** for common **cathode**

Fig.8 The experiment circuit

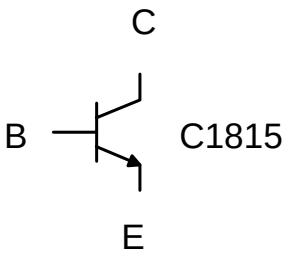
4. Components and its pin map

4.1 Opto Sensor

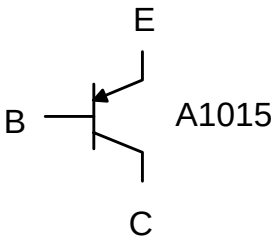




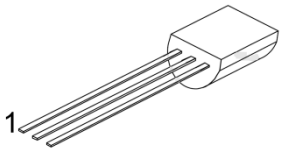
4.2 Bipolar Junction Transistor (BJT) PNP A1015 and NPN A1815



NPN A1815

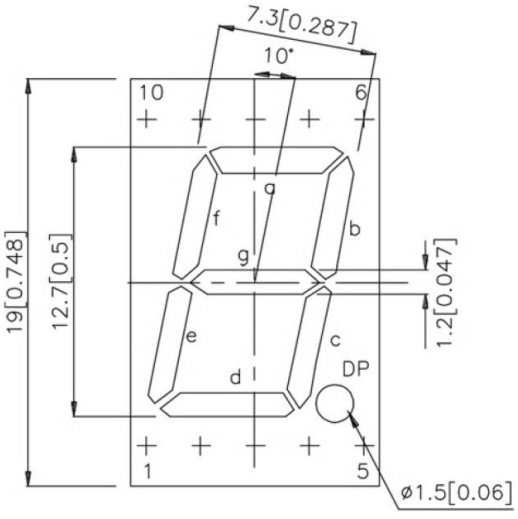


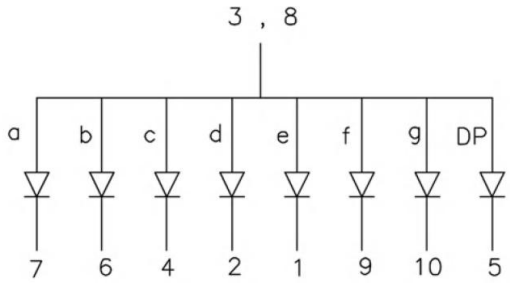
PNP A1015

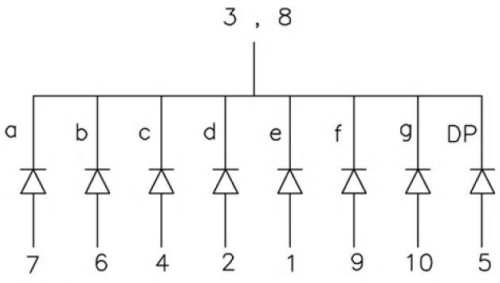


Pin Assignment		
1	2	3
E	C	B

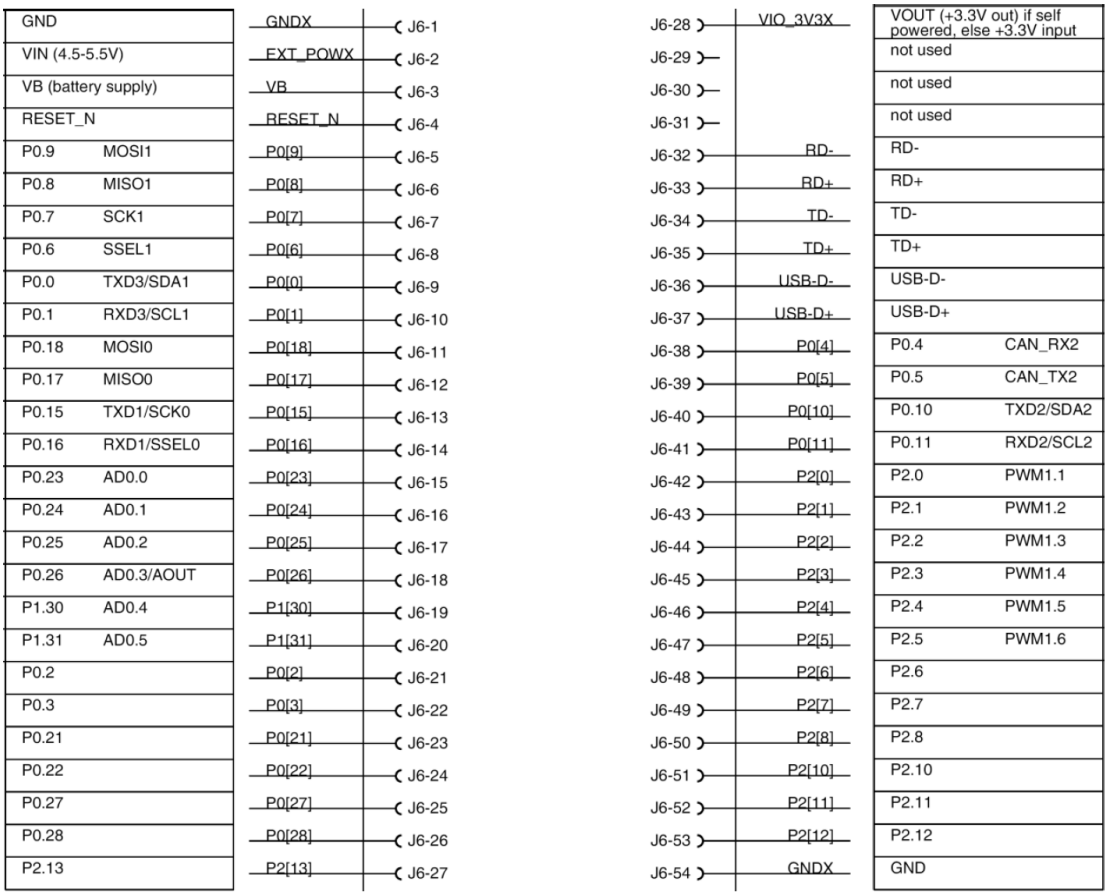
4.3 Seven-segment display

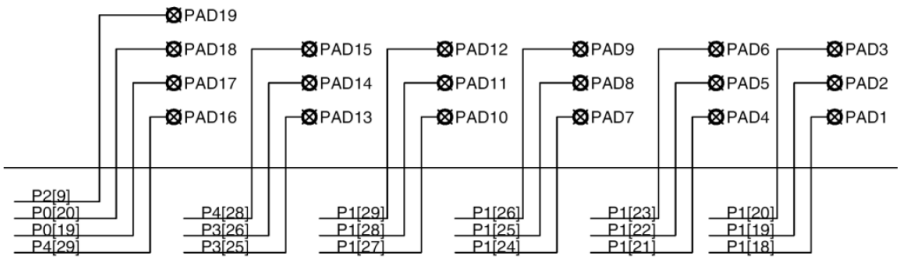






4.4 LPCXpresso pin map





5. References

[1] Rob Toulson and Tim Wilmshurst, "**Fast and Effective Embedded Systems Design Applying the ARM mbed**", Elsevier 2012.

[2] UM10360 LPC176x/5x User manual, NXP, April 2014