



Figure 1: ตัวอย่างของจอแสดงผล LED แบบ 7 ซีต

### วัตถุประสงค์ของการทดลองปฏิบัติการ

1. แสดงหลักการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และจอแสดงผล LED แบบ 7 ซีต
2. ปฏิบัติการทดลองใช้งานจอแสดงผล LED แบบ 7 ซีต

### 1 ทฤษฎีที่เกี่ยวข้อง

จอแสดงผล LED แบบ 7 ซีตดังแสดงในรูปที่ 1 เป็นจอแสดงผลที่มีการใช้งานมาอย่างยาวนาน จอแสดงผลแบบนี้สามารถผลเป็นตัวเลขได้เป็นอย่างดีแต่ก็สามารถแสดงผลเป็นตัวอักษรได้อย่างจำกัดโดยอาศัยการตัดแปลงเล็กน้อย โดยทั่วไปแล้วการเชื่อมโยงจอแสดงผลแบบนี้กับไมโครคอนโทรลเลอร์หรือระบบดิจิทัล สามารถทำได้โดยการเชื่อมต่อโดยตรงผ่านตัวต้านทานจำกัดกระแสโดยมีเงื่อนไขว่า Output port ของไมโครคอนโทรลเลอร์หรือระบบดิจิทัลจะต้องมีความสามารถในการขับกระแสเอาต์พุตได้ 15-20 mA เป็นอย่างต่ำ ด้วยเหตุผลนี้ทำให้ต้องใช้พลังงานไฟฟ้าค่อนข้างสูงในการควบคุมจอแสดงผลแบบนี้ เราสามารถลดการใช้พลังงานไฟฟ้าในการควบคุมจอแสดงผลนี้โดยใช้หลักการ Multiplexing (จะไม่นำเสนอในที่นี้) ในการใช้งานทั่วไปจะมีจอแสดงผล LED แบบ 7 ซีตอยู่สองชนิดคือ แบบขาแคโทดร่วม (common cathode type) และแบบขานอโดร่วม (common anode type) ดังแสดงในรูปที่ 2 และ 3

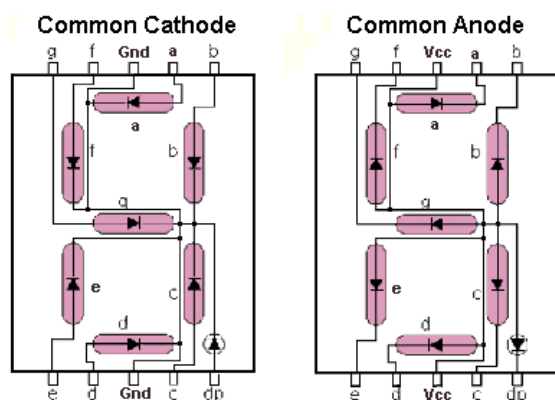


Figure 2: จอแสดงผล LED แบบแคโทดร่วมและแบบอานโดร่วม

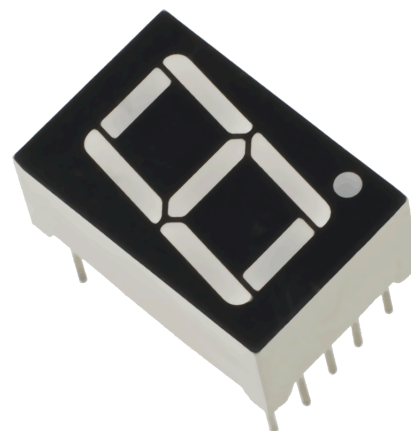


Figure 3: ตัวอย่างของจอแสดงผล LED แบบ 7 ซีต

การเชื่อมต่อกับจอแสดงผล LED แบบ 7 ซีต (แบบขาแคโทดร่วม common cathode) สามารถทำได้โดยการควบคุมการเปิดและปิดของหลอด LED ในแต่ละซีตดังแสดงด้วยตารางที่จริง Truth table ในรูปที่ 4

Digit Shown	Illuminated Segment (1 = illumination)						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Figure 4: ตารางตรรกะที่แท้จริง Truth Table สำหรับควบคุมการเปิดและปิดของจอแสดงผลแบบ 7 ซีด

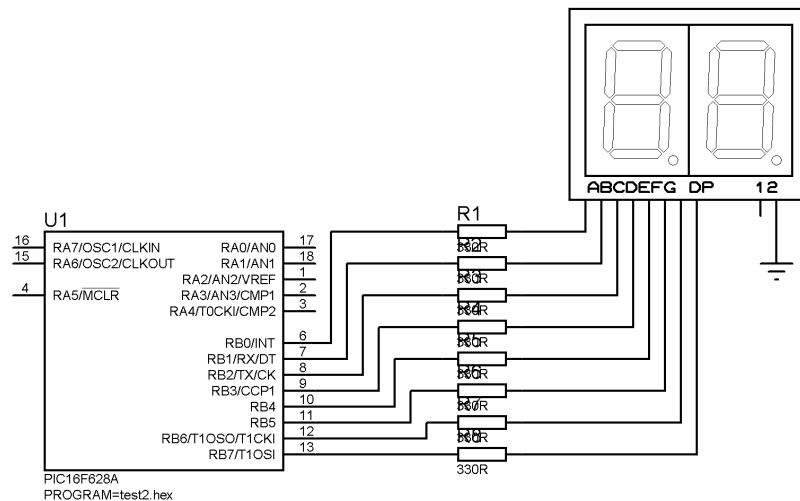


Figure 5: ตัวอย่างการเชื่อมต่อระหว่าง PIC16F628A และจอแสดงผลแบบ 7 ซีด

รูปที่ 5 แสดงวงจรการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ PIC16F628A และจอแสดงผล LED 7 ซีดแบบแคโทดร่วม ในวงจรนี้จะเห็นว่า ตัวต้านทาน 330 โอห์มจำนวนแปดตัวถูกใช้เพื่อจำกัดกระแสที่ไหลผ่านแต่ละซีดของจอแสดงผลให้อยู่ในปริมาณที่เหมาะสม โปรแกรมภาษาแอสเซมบลีและโฟลวชาตต่อไปนี้จะควบคุมให้จอแสดงผลแสดงตัวเลขจาก 0-9 และตัวอักษร A, B, C, D, E, F และจุดทศนิยม (dot point) ตัวเลขและตัวอักษรเหล่านี้ถูกแสดงเป็นเวลา 1 วินาทีโดยใช้โปรแกรมย่อยหน่วงเวลาที่ได้ทดลองใช้งานกันมาแล้วในการทดลองก่อนหน้านี้

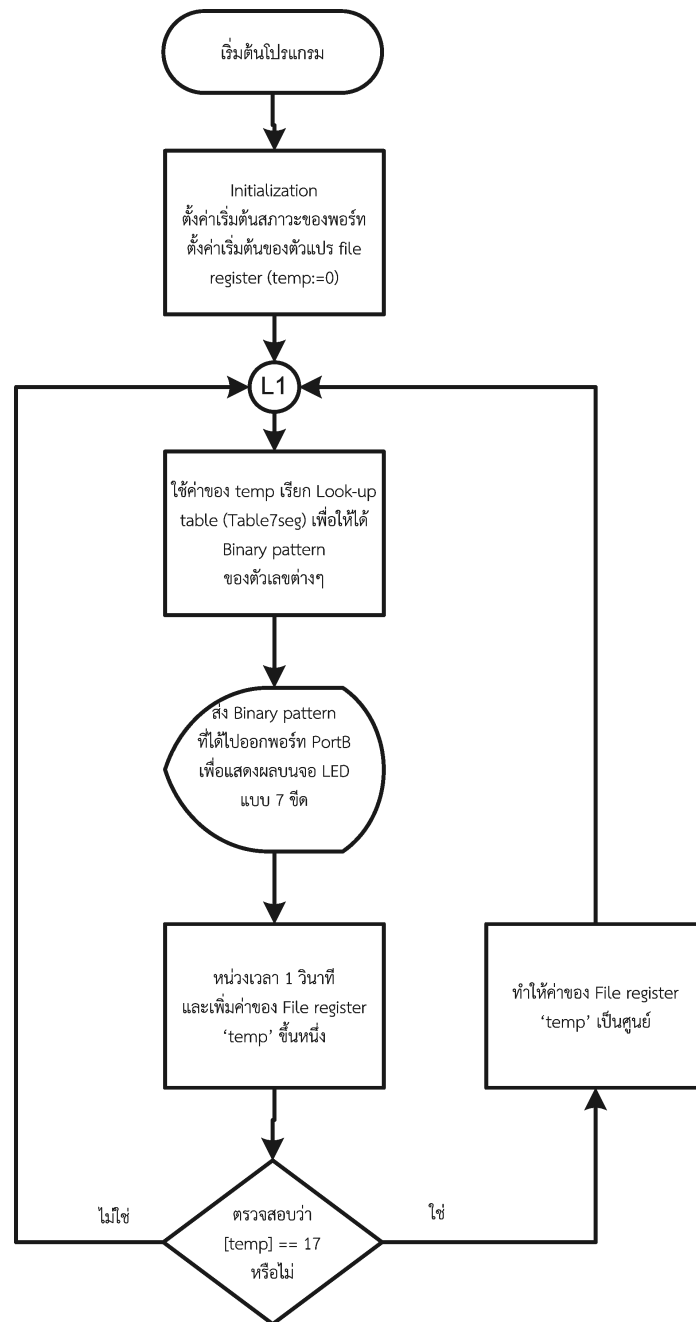


Figure 6: Flow Chart แสดงการควบคุมจอแสดงผล LED แบบ 7 ซิต

EGEE 380 Microprocessor	<h2 style="margin: 0;">Laboratory #5</h2> <h3 style="margin: 0;">7 Segment LED Interfacing</h3>	<div style="border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;">4</div> <div style="text-align: center;">8</div> </div>
Name _____ ID _____		
<pre> 1  ;***** 2  ;**** Program description:  Controlling the common cathode 7 seg LED ***** 3  ;**** by using the look-up table (Table7seg). The LED display will show ***** 4  ;**** numbers from 0-9, then characters 'A'-'F', '.' (dot-point) ***** 5  ;**** Programmer: ***** 6  ;***** 7      PROCESSOR PIC16F628 8      #include &lt;P16F628.INC&gt; 9      __CONFIG      _CP_OFF &amp; _MCLRE_OFF &amp; _INTRC_OSC_NOCLKOUT &amp; _LVP_OFF &amp;           _WDT_OFF 10 ;***** Variable Declaration (general purpose registers) 11         cblock    0x20 12             temp 13             temp1 14             count 15             count0 16             count1 17             count2 18         endc 19 20         ORG        0x00          ; Reset Vector 21         goto       Mymain        ; vector to main program 22 23         ORG        0x04          ; Interrupt Vector 24         goto       MyISR         ; vector to interrup service routine 25 26 MyISR:                                ; blank ISR routine 27         nop 28         retfie 29 30 Mymain:                                ; Main program begins here 31 32         call       Init          ; Call initialization subroutine 33 34         clrf       PORTB         ; Clear all used register files 35         bsf        PORTB,0 36         clrf       temp 37 38 L1: 39         movf       temp,w         ; Use [temp] to call 'Table7seg' 40         call       Table7seg 41         movwf      PORTB         ; Send the obtained 7 seg pattern to PORTB 42 43         movlw      .1             ; Delay for 1 s 44         call       DelayS 45 46         incf       temp,f         ; [temp] := [temp]+1 47 48         movlw      .17 49         subwf      temp,w 50         btfs       STATUS,Z       ; Check if temp=17 ? 51                                     ; (we want to display a total of 16 patterns) 52         goto       L1            ; No, go back and do it again 53 54         clrf       temp          ; Yes, clear 'temp' back to zero 55         goto       L1            ; Repeat the infinite loop 56 57 ; Look-up table for 7 Segment LED Patterns 58 Table7seg: 59         addwf      PCL, F 60         Segments  .GFEDCBA </pre>		
7 Segment LED Interfacing	Score _____	Date ____/____/____

EGEE 380 Microprocessor	<div>           Laboratory #5            7 Segment LED Interfacing         </div> <div>           Name _____ ID _____         </div>	<div>5</div> <div>8</div>
<div>61</div> <div>62</div> <div>63</div> <div>64</div> <div>65</div> <div>66</div> <div>67</div> <div>68</div> <div>69</div> <div>70</div> <div>71</div> <div>72</div> <div>73</div> <div>74</div> <div>75</div> <div>76</div> <div>77</div> <div>78</div> <div>79</div> <div>80</div> <div>81</div> <div>82</div> <div>83</div> <div>84</div> <div>85</div> <div>86</div> <div>87</div> <div>88</div> <div>89</div> <div>90</div> <div>91</div> <div>92</div> <div>93</div> <div>94</div> <div>95</div> <div>96</div> <div>97</div> <div>98</div> <div>99</div> <div>100</div> <div>101</div> <div>102</div> <div>103</div> <div>104</div> <div>105</div> <div>106</div> <div>107</div> <div>108</div> <div>109</div> <div>110</div> <div>111</div> <div>112</div> <div>113</div> <div>114</div> <div>115</div> <div>116</div> <div>117</div> <div>118</div> <div>119</div> <div>120</div> <div>121</div> <div>122</div>	<pre> retlw    B'00111111'      ; Number 0 retlw    B'00000110'      ; Number 1 retlw    B'01011011'      ; Number 2 retlw    B'01001111'      ; Number 3 retlw    B'01100110'      ; Number 4 retlw    B'01101101'      ; Number 5 retlw    B'01111101'      ; Number 6 retlw    B'00000111'      ; Number 7 retlw    B'01111111'      ; Number 8 retlw    B'01101111'      ; Number 9 retlw    B'01110111'      ; A retlw    B'01111100'      ; B retlw    B'00111001'      ; C retlw    B'01011110'      ; D retlw    B'01111001'      ; E retlw    B'01110001'      ; F retlw    B'10000000'      ; dot-point  ;===== ;* Delay 2 Routine - Decrement delay loop in Milisecond ;* 1 instruction cycle is 1 micro-second ;* at 4 Mhz X'tal frequency, 1MS = 1000 uS = 100x10 ;* where 100 iterations for inner loops, 10 iterations for ;* outter loops ;===== DelayMS:     movwf    count2     incf     count2,f     decfsz   count2,f     goto     \$+2     goto     \$+3     call     Delay1MS     goto     \$-4     return  Delay1MS:     movlw    .50           ; 1 cyc     movwf    count1        ; 1 cyc outterloop:     movlw    .5             ; 1 cyc * count1     nop      ; 1 cyc * count1     movwf    count0         ; 1 cyc * count1 innerloop:     decfsz   count0,F       ; 1 cyc * count1 * count0     goto     innerloop      ; 2 cyc * count1 * count0     decfsz   count1,F       ; 1 cyc * count1     goto     outterloop     ; 2 cyc * count1     return                ; 1 cyc ; total = 3 + (6+3.count0).count1 ; count0 = 5 , count1 = 50, total = 1053 cyc ??  ;***** Time delay subroutine for 1*[W] seconds by calling DelayMS subroutine DelayS:     movwf    temp1 delays_1:     movlw    .250     call     DelayMS     movlw    .250     call     DelayMS     movlw    .250     call     DelayMS     movlw    .250 </pre>	
7 Segment LED Interfacing	Score _____	Date ____/____/____

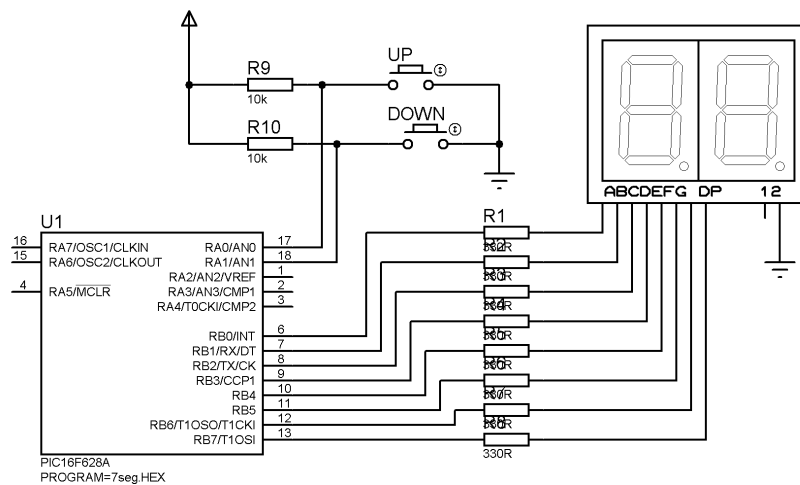


Figure 7: วงจรควบคุมจอแสดงผล LED แบบ 7 ซิตโดยใช้สวิตช์สองตัว UP, DOWN

```
123      call    DelayMS
124      decfsz  temp1,f
125      goto    delays_1
126      return
127
128  Init:      movlw  .7
129      banksel CMCON
130      movwf   CMCON          ; Disable analog comparator
131      banksel TRISB
132      movlw  0x00
133      movwf   TRISB          ; All pins of PortB are output pins
134      banksel PORTB
135      return
136
137      END
```

## 2 การทดลองปฏิบัติการ

เขียนโปรแกรมภาษาแอสเซมบลีเพื่อควบคุมวงจรในรูปที่ 7 โดยกำหนดให้ เมื่อกดสวิตช์ UP ให้ตัวเลขที่แสดงบนจอ LED แบบ 7 ซิตมีค่าเพิ่มขึ้นหนึ่งเช่น สมมุติว่าในขณะนั้นจอแสดงตัวเลข 5 เมื่อกดปุ่ม UP ให้จอแสดงตัวเลข 6 เป็นต้น และเมื่อกดสวิตช์ DOWN ให้ตัวเลขที่แสดงบนจอ LED แบบ 7 ซิตมีค่าลดลงหนึ่งเช่น สมมุติว่าในขณะนั้นจอแสดงตัวเลข 5 เมื่อกดปุ่ม DOWN ให้จอแสดงตัวเลข 4 เป็นต้น เพื่อให้ได้คะแนนเต็ม นักศึกษาจะต้อง

- เขียนโปรแกรมแอสเซมบลีในพื้นที่ว่างต่อไปนี้ (เขียนโฟลลวชาติประกอบถ้าเป็นไปได้) ให้เขียนโปรแกรมเฉพาะในส่วนของโปรแกรมหลักไม่จำเป็นต้องเขียนโปรแกรมน้อย
- ทำการจำลองการทำงานของโปรแกรมบน Proteus ISIS
- ทำการดาวน์โหลดโปรแกรมที่ได้ลงสู่บอร์ด PCK-1000 ต่อวงจรจาก PortA (RA0 (UP), RA1 (DOWN)) ไปที่สวิตช์จำนวนสองตัว (สวิตช์อยู่บนบอร์ด PCK-1000) ต่อวงจรจาก PortB (RB0-RB7) เข้ากับขั้วต่อของจอแสดงผล LED แบบ 7 ซิต (a-dp) ดังแสดงในรูปที่ 8 และสาธิตการทำงานของวงจรจริงให้กับอาจารย์ผู้สอน

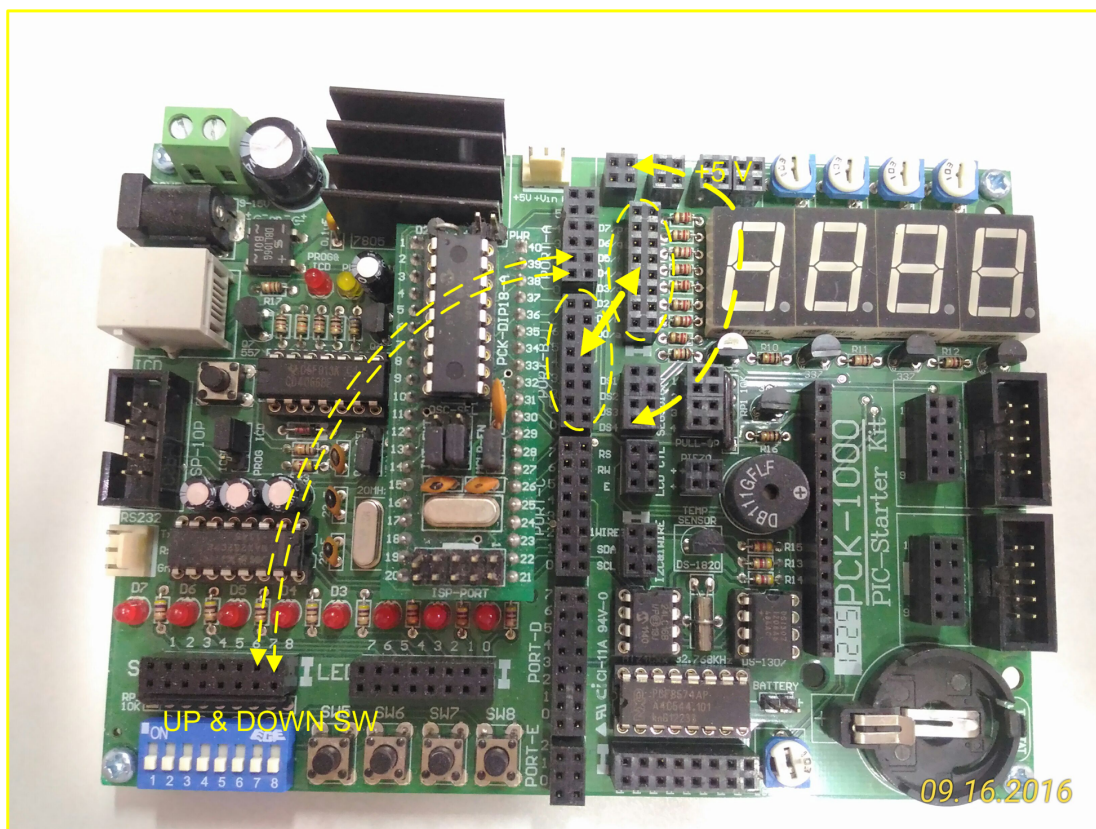


Figure 8: การต่อวงจรสำหรับการทดลองบนบอร์ด PCK-1000

(เขียนโปรแกรมในพื้นที่ว่างต่อไปนี้)

EGEE 380  
Microprocessor

Laboratory #5  
7 Segment LED Interfacing

Name \_\_\_\_\_ ID \_\_\_\_\_

8

8

(เขียนโปรแกรมในพื้นที่ว่างต่อไปนี้)