

S.C.ELECTRONICS

<http://www.sce-projects.com>

โครงการ

โมดูลแสดงผลและส่งข้อมูลสำหรับ MC 145151-2 PPL Frequency Synthesizer

โดย นายคมสันต์ ปีกะโต นักศึกษา วิศวกรรมคอมพิวเตอร์ (นักศึกษาฝึกงาน)

หลายคนคงปฏิเสธไม่ได้ว่าไมโครคอนโทรลเลอร์ ได้เข้ามามีบทบาทมากขึ้นในการประยุกต์ใช้งานแทนที่ระบบ อนาล็อก แบบเก่า เช่นในเครื่องเสียง เครื่องรับ ส่ง วิทยุ ทั้งนี้เพื่ออำนวยความสะดวกต่อผู้ใช้งานนั่นเอง ในโครงการนี้ เป็นการนำเอาไมโครคอนโทรลเลอร์มาประยุกต์ใช้งานร่วมกับ ไอซีเฟสล็อกแบบขนานเบอร์ MC145151-2 เพื่อแสดงผลและส่งข้อมูล โดยได้ยกตัวอย่าง การประยุกต์การใช้ไมโครคอนโทรลเลอร์ร่วมกับไอซีเบอร์ดังกล่าว ที่ใช้งานอยู่ในเครื่องส่งเอฟเอ็ม สเตอริโอมีดติเพล็กซ์

ในโครงการนี้เลือกใช้ไมโครคอนโทรลเลอร์เบอร์ PIC16F877 ของบริษัทไมโครชิพในการออกแบบ เพราะเป็นเบอร์ที่หาได้ง่ายมีพอร์ตใช้งานมาก และมีหนังสือภาษาไทยที่ยกตัวอย่างการใช้งานพิมพ์หลายเล่ม โดยจะขออนุญาตไม่กล่าวรายละเอียดในส่วนของวงจรเครื่องส่งวิทยุ ในเครื่องส่งที่เป็นต้นแบบก่อนที่จะนำไมโครคอนโทรลเลอร์เข้ามาทำงานร่วมนั้นอาศัยการปรับดิพลวิตซ์เพื่อกำหนดความถี่ของวงจรรอสซีสเลเตอร์แต่เราจะใช้ไมโครคอนโทรลเลอร์มาทำหน้าที่แทนดิพลวิตซ์และแสดงผลค่าความถี่ออกทาง 7Segment แทนก่อนอื่นมาดูตารางที่ใช้ปรับดิพลวิตซ์ก่อนเพราะต้องใช้ออกแบบโปรแกรมการทำงาน

Frequency (MHz)	Binary											Decimall
	10	9	8	7	6	5	4	3	2	1	0	
88.0	0	1	1	0	1	1	1	0	0	0	0	880
88.1	0	1	1	0	1	1	1	0	0	0	1	881
88.2	0	1	1	0	1	1	1	0	0	1	0	882
88.3	0	1	1	0	1	1	1	0	0	1	1	883
88.4	0	1	1	0	1	1	1	0	1	0	0	884
88.5	0	1	1	0	1	1	1	0	1	0	1	885
88.6	0	1	1	0	1	1	1	0	1	1	0	886
88.7	0	1	1	0	1	1	1	0	1	1	1	887
88.8	0	1	1	0	1	1	1	1	0	0	0	888
88.9	0	1	1	0	1	1	1	1	0	0	1	889
89.0	0	1	1	0	1	1	1	1	0	1	0	890
89.1	0	1	1	0	1	1	1	1	0	1	1	891
89.2	0	1	1	0	1	1	1	1	1	0	0	892
89.3	0	1	1	0	1	1	1	1	1	0	1	893
89.4	0	1	1	0	1	1	1	1	1	1	0	894
89.5	0	1	1	0	1	1	1	1	1	1	1	895
89.6	0	1	1	1	0	0	0	0	0	0	0	896
89.7	0	1	1	1	0	0	0	0	0	0	1	897
89.8	0	1	1	1	0	0	0	0	0	1	0	898
89.9	0	1	1	1	0	0	0	0	0	1	1	899
90.0	0	1	1	1	0	0	0	0	1	0	0	900
90.1	0	1	1	1	0	0	0	0	1	0	1	901
90.2	0	1	1	1	0	0	0	0	1	1	0	902
90.3	0	1	1	1	0	0	0	0	1	1	1	903
90.4	0	1	1	1	0	0	0	1	0	0	0	904
90.5	0	1	1	1	0	0	0	1	0	0	1	905
90.6	0	1	1	1	0	0	0	1	0	1	0	906
90.7	0	1	1	1	0	0	0	1	0	1	1	907
90.8	0	1	1	1	0	0	0	1	1	0	0	908
90.9	0	1	1	1	0	0	0	1	1	0	1	909
91.0	0	1	1	1	0	0	0	1	1	1	0	910
91.1	0	1	1	1	0	0	0	1	1	1	1	911
91.2	0	1	1	1	0	0	1	0	0	0	0	912
91.3	0	1	1	1	0	0	1	0	0	0	1	913
91.4	0	1	1	1	0	0	1	0	0	1	0	914
91.5	0	1	1	1	0	0	1	0	0	1	1	915
91.6	0	1	1	1	0	0	1	0	1	0	0	916
91.7	0	1	1	1	0	0	1	0	1	0	1	917
91.8	0	1	1	1	0	0	1	0	1	1	0	918

91.9	0	1	1	1	0	0	1	0	1	1	1	919
92.0	0	1	1	1	0	0	1	1	0	0	0	920
92.1	0	1	1	1	0	0	1	1	0	0	1	921
92.2	0	1	1	1	0	0	1	1	0	1	0	922
92.3	0	1	1	1	0	0	1	1	0	1	1	923
92.4	0	1	1	1	0	0	1	1	1	0	0	924
92.5	0	1	1	1	0	0	1	1	1	0	1	925
92.6	0	1	1	1	0	0	1	1	1	1	0	926
92.7	0	1	1	1	0	0	1	1	1	1	1	927
92.8	0	1	1	1	0	1	0	0	0	0	0	928
92.9	0	1	1	1	0	1	0	0	0	0	1	929
93.0	0	1	1	1	0	1	0	0	0	1	0	930
93.1	0	1	1	1	0	1	0	0	0	1	1	931
93.2	0	1	1	1	0	1	0	0	1	0	0	932
93.3	0	1	1	1	0	1	0	0	1	0	1	933
93.4	0	1	1	1	0	1	0	0	1	1	0	934
93.5	0	1	1	1	0	1	0	0	1	1	1	935
93.6	0	1	1	1	0	1	0	1	0	0	0	936
93.7	0	1	1	1	0	1	0	1	0	0	1	937
93.8	0	1	1	1	0	1	0	1	0	1	0	938
93.9	0	1	1	1	0	1	0	1	0	1	1	939
94.0	0	1	1	1	0	1	0	1	1	0	0	940
94.1	0	1	1	1	0	1	0	1	1	0	1	941
94.2	0	1	1	1	0	1	0	1	1	1	0	942
94.3	0	1	1	1	0	1	0	1	1	1	1	943
94.4	0	1	1	1	0	1	1	0	0	0	0	944
94.5	0	1	1	1	0	1	1	0	0	0	1	945
94.6	0	1	1	1	0	1	1	0	0	1	0	946
94.7	0	1	1	1	0	1	1	0	0	1	1	947
94.8	0	1	1	1	0	1	1	0	1	0	0	948
94.9	0	1	1	1	0	1	1	0	1	0	1	949
95.0	0	1	1	1	0	1	1	0	1	1	0	950
95.1	0	1	1	1	0	1	1	0	1	1	1	951
95.2	0	1	1	1	0	1	1	1	0	0	0	952
95.3	0	1	1	1	0	1	1	1	0	0	1	953
95.4	0	1	1	1	0	1	1	1	0	1	0	954
95.5	0	1	1	1	0	1	1	1	0	1	1	955
95.6	0	1	1	1	0	1	1	1	1	0	0	956
95.7	0	1	1	1	0	1	1	1	1	0	1	957
95.8	0	1	1	1	0	1	1	1	1	1	0	958
95.9	0	1	1	1	0	1	1	1	1	1	1	959
96.0	0	1	1	1	1	0	0	0	0	0	0	960
96.1	0	1	1	1	1	0	0	0	0	0	1	961
96.2	0	1	1	1	1	0	0	0	0	1	0	962
96.3	0	1	1	1	1	0	0	0	0	1	1	963
96.4	0	1	1	1	1	0	0	0	1	0	0	964
96.5	0	1	1	1	1	0	0	0	1	0	1	965
96.6	0	1	1	1	1	0	0	0	1	1	0	966
96.7	0	1	1	1	1	0	0	0	1	1	1	967
96.8	0	1	1	1	1	0	0	1	0	0	0	968
96.9	0	1	1	1	1	0	0	1	0	0	1	969
97.0	0	1	1	1	1	0	0	1	0	1	0	970
97.1	0	1	1	1	1	0	0	1	0	1	1	971
97.2	0	1	1	1	1	0	0	1	1	0	0	972
97.3	0	1	1	1	1	0	0	1	1	0	1	973
97.4	0	1	1	1	1	0	0	1	1	1	0	974
97.5	0	1	1	1	1	0	0	1	1	1	1	975
97.6	0	1	1	1	1	0	1	0	0	0	0	976
97.7	0	1	1	1	1	0	1	0	0	0	1	977
97.8	0	1	1	1	1	0	1	0	0	1	0	978
97.9	0	1	1	1	1	0	1	0	0	1	1	979
98.0	0	1	1	1	1	0	1	0	1	0	0	980
98.1	0	1	1	1	1	0	1	0	1	0	1	981
98.2	0	1	1	1	1	0	1	0	1	1	0	982
98.3	0	1	1	1	1	0	1	0	1	1	1	983
98.4	0	1	1	1	1	0	1	1	0	0	0	984
98.5	0	1	1	1	1	0	1	1	0	0	1	985

98.6	0	1	1	1	1	0	1	1	0	1	0	986
98.7	0	1	1	1	1	0	1	1	0	1	1	987
98.8	0	1	1	1	1	0	1	1	1	0	0	988
98.9	0	1	1	1	1	0	1	1	1	0	1	989
99.0	0	1	1	1	1	0	1	1	1	1	0	990
99.1	0	1	1	1	1	0	1	1	1	1	1	991
99.2	0	1	1	1	1	1	0	0	0	0	0	992
99.3	0	1	1	1	1	1	0	0	0	0	1	993
99.4	0	1	1	1	1	1	0	0	0	0	1	994
99.5	0	1	1	1	1	1	0	0	0	1	1	995
99.6	0	1	1	1	1	1	0	0	1	0	0	996
99.7	0	1	1	1	1	1	0	0	1	0	1	997
99.8	0	1	1	1	1	1	0	0	1	1	0	998
99.9	0	1	1	1	1	1	0	0	1	1	1	999
100.0	0	1	1	1	1	1	0	1	0	0	0	1000
100.1	0	1	1	1	1	1	0	1	0	0	1	1001
100.2	0	1	1	1	1	1	0	1	0	1	0	1002
100.3	0	1	1	1	1	1	0	1	0	1	1	1003
100.4	0	1	1	1	1	1	0	1	1	0	0	1004
100.5	0	1	1	1	1	1	0	1	1	0	1	1005
100.6	0	1	1	1	1	1	0	1	1	1	0	1006
100.7	0	1	1	1	1	1	0	1	1	1	1	1007
100.8	0	1	1	1	1	1	1	0	0	0	0	1008
100.9	0	1	1	1	1	1	1	0	0	0	1	1009
101.0	0	1	1	1	1	1	1	0	0	1	0	1010
101.1	0	1	1	1	1	1	1	0	0	1	1	1011
101.2	0	1	1	1	1	1	1	0	1	0	0	1012
101.3	0	1	1	1	1	1	1	0	1	0	1	1013
101.4	0	1	1	1	1	1	1	0	1	1	0	1014
101.5	0	1	1	1	1	1	1	0	1	1	1	1015
101.6	0	1	1	1	1	1	1	1	0	0	0	1016
101.7	0	1	1	1	1	1	1	1	0	0	1	1017
101.8	0	1	1	1	1	1	1	1	0	1	0	1018
101.9	0	1	1	1	1	1	1	1	0	1	1	1019
102.0	0	1	1	1	1	1	1	1	1	0	0	1020
102.1	0	1	1	1	1	1	1	1	1	0	1	1021
102.2	0	1	1	1	1	1	1	1	1	1	0	1022
102.3	0	1	1	1	1	1	1	1	1	1	1	1023
102.4	1	0	0	0	0	0	0	0	0	0	0	1024
102.5	1	0	0	0	0	0	0	0	0	0	1	1025
102.6	1	0	0	0	0	0	0	0	0	0	1	1026
102.7	1	0	0	0	0	0	0	0	0	0	1	1027
102.8	1	0	0	0	0	0	0	0	0	1	0	1028
102.9	1	0	0	0	0	0	0	0	0	1	0	1029
103.0	1	0	0	0	0	0	0	0	0	1	1	1030
103.1	1	0	0	0	0	0	0	0	0	1	1	1031
103.2	1	0	0	0	0	0	0	0	1	0	0	1032
103.3	1	0	0	0	0	0	0	0	1	0	0	1033
103.4	1	0	0	0	0	0	0	0	1	0	1	1034
103.5	1	0	0	0	0	0	0	0	1	0	1	1035
103.6	1	0	0	0	0	0	0	0	1	1	0	1036
103.7	1	0	0	0	0	0	0	0	1	1	0	1037
103.8	1	0	0	0	0	0	0	0	1	1	1	1038
103.9	1	0	0	0	0	0	0	0	1	1	1	1039
104.0	1	0	0	0	0	0	0	1	0	0	0	1040
104.1	1	0	0	0	0	0	0	1	0	0	0	1041
104.2	1	0	0	0	0	0	0	1	0	0	1	1042
104.3	1	0	0	0	0	0	0	1	0	0	1	1043
104.4	1	0	0	0	0	0	0	1	0	1	0	1044
104.5	1	0	0	0	0	0	0	1	0	1	0	1045
104.6	1	0	0	0	0	0	0	1	0	1	1	1046
104.7	1	0	0	0	0	0	0	1	0	1	1	1047
104.8	1	0	0	0	0	0	0	1	1	0	0	1048
104.9	1	0	0	0	0	0	0	1	1	0	0	1049
105.0	1	0	0	0	0	0	0	1	1	0	1	1050

105.1	1	0	0	0	0	0	1	1	0	1	1	1051
105.2	1	0	0	0	0	0	1	1	1	0	0	1052
105.3	1	0	0	0	0	0	1	1	1	0	1	1053
105.4	1	0	0	0	0	0	1	1	1	1	0	1054
105.5	1	0	0	0	0	0	1	1	1	1	1	1055
105.6	1	0	0	0	0	1	0	0	0	0	0	1056
105.7	1	0	0	0	0	1	0	0	0	0	1	1057
105.8	1	0	0	0	0	1	0	0	0	1	0	1058
105.9	1	0	0	0	0	1	0	0	0	1	1	1059
106.0	1	0	0	0	0	1	0	0	1	0	0	1060
106.1	1	0	0	0	0	1	0	0	1	0	1	1061
106.2	1	0	0	0	0	1	0	0	1	1	0	1062
106.3	1	0	0	0	0	1	0	0	1	1	1	1063
106.4	1	0	0	0	0	1	0	1	0	0	0	1064
106.5	1	0	0	0	0	1	0	1	0	0	1	1065
106.6	1	0	0	0	0	1	0	1	0	1	0	1066
106.7	1	0	0	0	0	1	0	1	0	1	1	1067
106.8	1	0	0	0	0	1	0	1	1	0	0	1068
106.9	1	0	0	0	0	1	0	1	1	0	1	1069
107.0	1	0	0	0	0	1	0	1	1	1	0	1070
107.1	1	0	0	0	0	1	0	1	1	1	1	1071
107.2	1	0	0	0	0	1	1	0	0	0	0	1072
107.3	1	0	0	0	0	1	1	0	0	0	1	1073
107.4	1	0	0	0	0	1	1	0	0	1	0	1074
107.5	1	0	0	0	0	1	1	0	0	1	1	1075
107.6	1	0	0	0	0	1	1	0	1	0	0	1076
107.7	1	0	0	0	0	1	1	0	1	0	1	1077
107.8	1	0	0	0	0	1	1	0	1	1	0	1078
107.9	1	0	0	0	0	1	1	0	1	1	1	1079
108.0	1	0	0	0	0	1	1	1	0	0	0	1080

ตารางที่ 1 ตารางการปรับดีฟลิวซ์

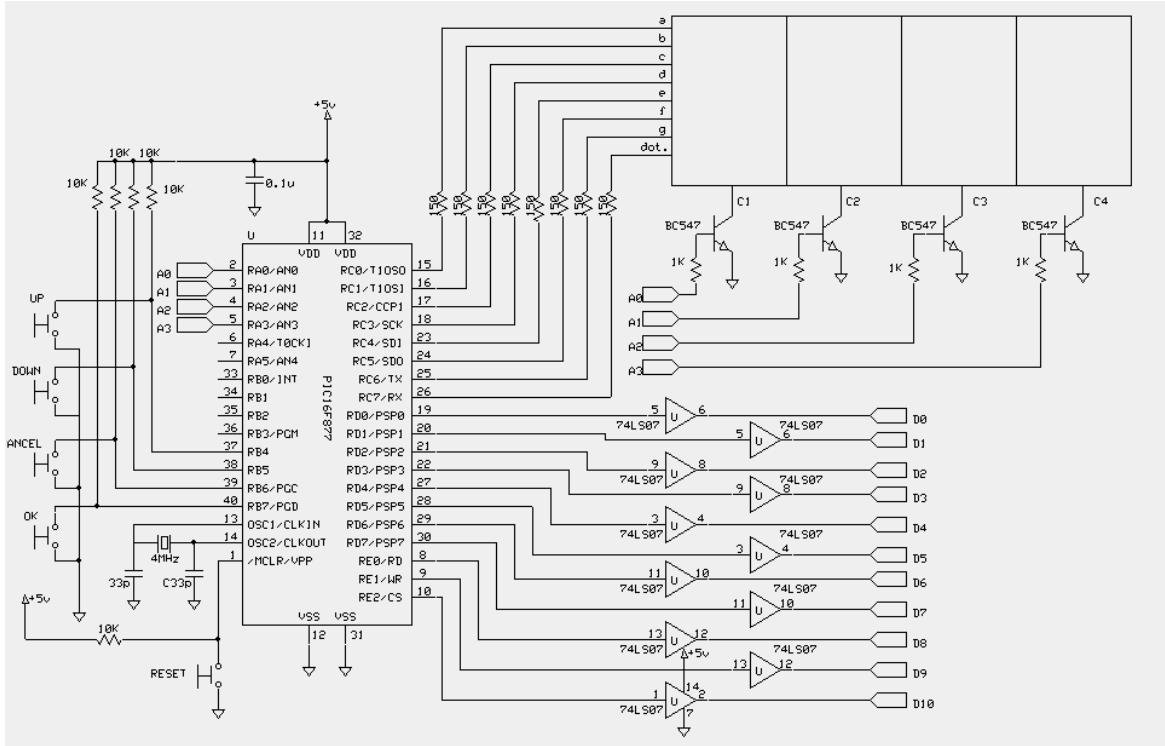
แนวคิดในการออกแบบเนื่องจากค่าเลขฐานสองและค่าความถี่ในตารางจะเห็นได้ว่ามีความสอดคล้องกันคือค่าของเลขฐานสองที่แปลงเป็นฐานสิบแล้วจะเท่ากับค่าความถี่โดยคิดให้เลขตัวท้ายสุดเป็นจุดทศนิยม เราจึงสามารถส่งข้อมูลที่ตรงกับค่าการแสดงผลทาง 7 Segment ให้ MC145151-2 ได้ทันที วงจรที่ใช้แน่นอนออกแบบมาให้ยึดหยุ่นต่อการเขียนโปรแกรมได้สะดวกโดยมีการใช้งานพอร์ตต่างๆดังนี้

- พอร์ต A0-A3 ใช้สำหรับเป็นตัวสแกน 7 Segment คอมมอนคาโทด
- พอร์ต B4-B7 ใช้เป็นอินพุตรับค่าจากสวิตซ์
- พอร์ต C0-C7 ใช้เป็นเอาต์พุตสำหรับ ขับ 7 Segment
- พอร์ต D0-D7 ใช้ เป็นเอาต์พุตสำหรับ บิต 0-7
- พอร์ต E0-E2 ใช้เป็นเอาต์พุตสำหรับ บิต 8-10

การทำงานของวงจร

ไมโครคอนโทรลเลอร์จะเป็นตัวจัดการทั้งหมดเพียงแต่ต่อพอร์ตอินพุต เอาต์พุตให้ถูกต้องหากจะเลือกใช้ตำแหน่งพอร์ตอื่นที่ไม่ตรงกับรูปวงจรก็สามารถทำได้แต่ต้องแก้ไขโปรแกรมใหม่และดูรายละเอียดของพอร์ตนั้นๆให้ดูว่าสามารถใช้งานแบบใดได้บ้าง โดยมีอุปกรณ์ที่ต่อรวมดังต่อไปนี้

7 Segment เป็นแบบคอมมอนคาโทด ใช้ทรานซิสเตอร์ชนิดเอ็นพีเอ็นเบอร์ BC547 ต่อเป็นสวิตซ์เพื่อลดการดึงกระแสจากพอร์ต A0-A3 แอลอีดีที่ป็นซีกเมนต์ รวมทั้ง dot (จุด) จะต่อร่วมกันทั้งหมดสี่ตัวโดยใช้การเขียนโปรแกรมสแกนให้ 7 Segment ติดที่ละหลักด้วยความเร็วสูงจนตาของคนเราจับการกะพริบไม่ทันในขณะที่การสแกนที่หลักใดเกิดขึ้นข้อมูลตัวเลขของหลักนั้นๆก็ถูกส่งออกมาทางพอร์ต C0-C7 เช่นกัน ทำให้ตัวเลขที่แสดงผลเป็นค่าประจำหลักเท่านั้น ในส่วนของอินพุตสวิตซ์มีปุ่มสำหรับใช้งานดังนี้ B4เป็นอินพุตสวิตซ์ขึ้นขึ้น(UP) B5 เป็นอินพุตสวิตซ์นับลง(DOWN) B6เป็นอินพุตสวิตซ์ยกเลิก(CANCEL) และ B7เป็นอินพุตสวิตซ์คำสั่งตกลง (OK) การโดยมีความต้านทานต่อพูล์อัพเพื่อให้โปรแกรมตรวจจับการกดสวิตซ์ หากมีการกดสวิตซ์ล่อจิกที่พอร์ตจะเท่ากับ"0"ในขณะที่ปล่อยสวิตซ์ล่อจิกที่พอร์ตจะเท่ากับ"1" ในส่วนของการส่งข้อมูลแบบขนานทางพอร์ต D0-D7 และ E0-E2 นั้นเลือกใช้อิซีทีทีแอลบัฟเฟอร์ไดร์เวอร์ เบอร์ 74LS07 เอาต์พุตเป็นแบบโอเพนคอลเล็กเตอร์ Open Collector เป็นตัวขับโดยการออกแบบให้พยายามหลีกเลี่ยงที่จะใช้งานพอร์ตโดยตรงในการเชื่อมต่อกับไอซีเฟลลิกลูปหึ่งนี้เพื่อสถียรภาพของไมโครคอนโทรลเลอร์เอง ในรูปวงจรถ้าตำแหน่งของขาไอซี74LS07 ในวงจรวงจรไม่ตรงกับลายทองแดงที่ออกแบบมาให้แต่เพียงให้เข้าใจว่าใช้บัฟเฟอร์เกดก่อนส่งข้อมูลออกเอาต์พุต



รูปที่ 1 วงจร

ในตัวอย่างโปรแกรมที่เขียนขึ้นเป็นภาษาซีใช้คอมไพเลอร์ CCS Compiler ในการพัฒนา การทำงานของโปรแกรมตัวอย่างนั้นเริ่มแรกโปรแกรมจะอ่านข้อมูลที่บันทึกไว้ล่าสุดจาก EEPROM ข้อมูลที่ได้ออกมาแสดงผลและส่งไปให้ไอซีเฟลลิลอูป หากมีการกดปุ่ม UP หรือ DOWN หน้าจอจะแสดงการนับความถี่ตามการกดปุ่มและกระพริบเป็นจังหวะ หากการกดสวิทซ์ค้างเป็นเวลานานประมาณ 3 วินาที การนับจะเป็นแบบเร็ว และค่าข้อมูลที่ส่งออกทางพอร์ต D0-D7 และ E0-E2 จะยังไม่มีการเปลี่ยนแปลงแต่อย่างใดจนกว่าจะมีการตกลง หรือกดสวิทซ์ OK และหากมีการกดตกลง ข้อมูลใหม่จะถูกเขียนลงใน EEPROM ด้วย หากในขณะที่ ข้อมูลที่แสดงผลบน 7 Segment ยังกระพริบอยู่หรือยังไม่มีการตกลงหรือกดสวิทซ์ OK หากกดปุ่มยกเลิก หรือ CANCEL หน้าจอจะกลับมาแสดงผลค่าความถี่ที่กำลังส่งออกปัจจุบัน หรือ ข้อมูลเดิมก่อนมีการเปลี่ยนแปลงจากการกดปุ่ม UP หรือ DOWN หากไม่มีการกดตกลง หรือยกเลิกและไม่มีการกดปุ่ม UP หรือ DOWN เมื่อเวลาผ่านไปประมาณ 10 วินาทีก็จะกลับมาแสดงผลความถี่และส่งข้อมูลเดิมก่อนมีการเปลี่ยนแปลงเช่นกัน

โปรแกรม

```
#include <16F877.h> // เรียก include ไฟล์ 16F877.h ใช้งาน
#define CLOCK_SP 400000 // คอนฟิกค่าความถี่ของคริสตัลให้ตรงกับค่าที่ใช้งาน
#fuses XT // เลือกโหมดของวงจรออสซิลเลเตอร์เป็น XT
#fuses NOLVP, NOWDT // ไม่ใช้การโปรแกรมโหมดแรงดันต่ำ, ไม่ใช้ Watchdogtimer
#fuses NOPROTECT // ไม่มีการป้องกันการเขียนหน่วยความจำโปรแกรม
#use delay (clock=CLOCK_SP) // เซ็ตค่าให้กับฟังก์ชัน delay_ms()
//สร้างตารางเก็บข้อมูลเพื่อแสดงผลให้กับ 7 Segments
char num_led[16] = {0x3F,0x06,0x5B,0x4F,0x66, // 0,1,2,3,4
                   0x6D,0x7D,0x07,0x7F,0x6F, // 5,6,7,8,9
                   0x77,0x7C,0x39,0x5E,0x79, // A,b,C,d,e
                   0x71,; // F

int16 counter; //สร้างตัวแปร int16 counter เพื่อเป็นตัวนับ
char Display_Buff[5]; //สร้างตารางเพื่อพักข้อมูลที่จะแสดงผลแต่ละหลัก
char Disp=0; //ตัวแปรส่งข้อมูลแสดงผล
int16 scan=0; //ตัวแปรกำหนดการสแกนในโปรแกรมย่อยแสดงผล
int Dot=3; //กำหนดตำแหน่งจุด Dot ที่จะแสดงผลตำแหน่งที่3
int Dark=0; //ตัวแปรเซ็ตค่าเพื่อกำหนดให้ 7 Segments ดับทั้งหมด
int chk1=0, chk2=0 , step=0, step2; //ตัวแปรตรวจเช็การกดสวิทซ์
int16 times=0, times2=0 ; //ตัวแปรตรวจเช็การกดสวิทซ์ค้าง
int16 sycle0=0, sycle1=100, Sec=0; //ตัวแปรกำหนดค่าของการกระพริบของตัวเลข
int pulse = 0; //ตัวแปรตรวจเช็คสถานะว่ามีการกดสวิทซ์แล้ว
int w = 1, r = 1; //ตัวแปรสั่งการเขียนและอ่านหน่วยความจำในครั้งแรกของโปรแกรม

//////////โปรแกรมย่อยสำหรับเขียนหน่วยความจำ EEPROM//////////
void WriteData_EEPROM(void);
void WriteData__EEPROM(void)
```

```

{
int16 BuffEEPROM[16]; //ตารางสำหรับพักข้อมูลที่จะเขียนลงหน่วยความจำ EEPROM
int i=0; //ตัวแปรสำหรับนับรอบลูปการเขียนข้อมูล
BuffEEPROM[0] = counter; //กำหนดให้ตารางตำแหน่งแรกมีค่าเท่ากับตัวแปรcounter
write_eeprom(0x00,BuffEEPROM[0]); //เขียนข้อมูลจากบัฟเฟอร์ตำแหน่งแรกลง eeprom ตำแหน่ง0x00
for(i=0x01;i<=0x0A;i++){ //กำหนดการวนลูป i
BuffEEPROM[i]=BuffEEPROM[i-1]/2; //เก็บข้อมูลจากตำแหน่งก่อนนี้หารสองลงตำแหน่งปัจจุบัน
write_eeprom(i,BuffEEPROM[i]); //เขียนข้อมูลจากบัฟเฟอร์ลงตำแหน่งหน่วยความจำตามลูป i
}
w = 0; // เคลียร์ ตัวแปร w = 0เมื่อเขียนข้อมูลลง eeprom แล้ว
}
//////////////////////////////////โปรแกรมย่อยสำหรับอ่านหน่วยความจำ EEPROM//////////////////////////////////
void ReadData_EEPROM(void);
void ReadData_EEPROM(void)
{
int16 DataRead = 0; //ตัวแปรเก็บข้อมูลเรียงแล้วที่อ่านจากอีพีรอม
int DataBuffEEPROM[16]; //ตารางพักข้อมูลที่อ่านจากอีพีรอม
int OutputBuff_D,OutputBuff_E; //ตัวแปรพักข้อมูลก่อนส่งออกพอร์ต D และ E
int j=0 ;
for(j=0;j<=10;j++){
DataBuffEEPROM[j]= read_eeprom(j); //อ่านข้อมูลจากอีพีรอมมาเก็บในตารางบัฟเฟอร์ชี้ตำแหน่งด้วยลูป
}
//แปลงข้อมูลในบัฟเฟอร์ 8บิตให้เก็บในตัวแปร DataRead 16บิต
if(DataBuffEEPROM[10]%2 == 1)DataRead = 1024;
if(DataBuffEEPROM[9]%2 == 1)DataRead = DataRead+512;
if(DataBuffEEPROM[8]%2 == 1)DataRead = DataRead+256;
if(DataBuffEEPROM[7]%2 == 1)DataRead = DataRead+128;
if(DataBuffEEPROM[6]%2 == 1)DataRead = DataRead+64;
if(DataBuffEEPROM[5]%2 == 1)DataRead = DataRead+32;
if(DataBuffEEPROM[4]%2 == 1)DataRead = DataRead+16;
if(DataBuffEEPROM[3]%2 == 1)DataRead = DataRead+8;
if(DataBuffEEPROM[2]%2 == 1)DataRead = DataRead+4;
if(DataBuffEEPROM[1]%2 == 1)DataRead = DataRead+2;
if(DataBuffEEPROM[0]%2 == 1)DataRead = DataRead+1;
counter = DataRead; //ให้ตัวแปร counter เท่ากับ DataRead
OutputBuff_D = read_eeprom(0); // อ่านค่าจากหน่วยความจำอีพีรอมตำแหน่ง 0 ลงOutputBuff_D
OutputBuff_E = read_eeprom(8); //อ่านค่าจากหน่วยความจำอีพีรอมตำแหน่ง 8 ลงOutputBuff_E
output_d(OutputBuff_D); //ส่งข้อมูลออกพอร์ต D
output_e(OutputBuff_E); //ส่งข้อมูลออกพอร์ต E
r = 0; // เคลียร์ ตัวแปร r = 0 เมื่ออ่านข้อมูลจากeeprom แล้ว
}//////////////////////////////////โปรแกรมย่อยแยกค่าตัวเลขประจำหลัก//////////////////////////////////
void Converse(void);
void Converse(void){
Display_Buff[0] = (counter/1000); //เก็บค่าตัวเลขหลักพันจาก counter ลง Display_Buff[0]
Display_Buff[1] = (counter%1000/100); //เก็บค่าตัวเลขหลักร้อยจาก counter ลง Display_Buff[1]
Display_Buff[2] = (counter%100)/10; //เก็บค่าตัวเลขหลักสิบจาก counter ลง Display_Buff[2]
Display_Buff[3] = counter%10; //เก็บค่าตัวเลขหลักหน่วยจาก counter ลง Display_Buff[3]
}
//////////////////////////////////โปรแกรมย่อยรับค่าจากสวิตซ์//////////////////////////////////
void Input_sw(void);
void Input_sw(void)
{
if(input(PIN_B4)==0 || input(PIN_B5)==0){ //เมื่อมีการกดสวิตซ์ นับขึ้นหรือนับลง
puse = 1; //เก็บค่าสถานะว่ามีการกดสวิตซ์
Sec = 0; //ตั้งค่าเริ่มต้นตัวนับเวลาการกระพริบ
}
if(input(PIN_B4)==1 && input(PIN_B5)==1 && puse==1){
if(sycle0<100 && sycle1 == 100){ //เมื่อปล่อยสวิตซ์แล้วและ puse = 1
sycle0 = sycle0+1; //เพิ่มค่าตัวแปร sycle0
Dark = 0; //การแสดงผลบน7 Segment เป็นไปตามปกติ
if(sycle0==100){ //ถ้า sycle0 = 100 //เป็นการกำหนดเวลาการติดของแอลอีดี7 Segment
sycle1 = 0; // ให้ sycle1 = 0
}
}
if(sycle1<100 && sycle0 == 100){ //
sycle1 = sycle1+1; //เพิ่มค่าตัวแปร sycle1
Dark = 1; //ให้7 Segment ดับทกอลลิมน์
if(sycle1 == 100){ //ถ้า sycle1 == 100//เป็นการกำหนดเวลาการดับของแอลอีดี7 Segment
sycle0 = 0; // ให้ sycle0 = 0
}
}
}
}

```

```

Sec = Sec+1;           // เพิ่มค่าในตัวแปร Sec
if(Sec==3000){        // ถ้าตัวแปร Sec เท่ากับ 3000 (ใช้เวลาประมาณ 10 วินาที)
  ReadData_EEPROM(); // เรียกโปรแกรมย่อยอ่านข้อมูลในอีพรอม
  puse = 0;           // เคลียร์ค่าตัวแปรตรวจสอบการกดสวิตช์นับขึ้นลง
  Sec = 0;            // เคลียร์ค่าตัวนับเวลาการกระพริบ
  Dark = 0;          // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segment
}

if(input(PIN_B7) == 0){ // หากมีการกดสวิตช์ตกลง OK SW_PIN_B7
  WriteData_EEPROM(); // เรียกโปรแกรมย่อยเขียนหน่วยความจำลงอีพรอม
  ReadData_EEPROM(); // เรียกโปรแกรมย่อยอ่านหน่วยความจำจากอีพรอม
  puse = 0;           // เคลียร์ค่าตัวแปรตรวจสอบการกดสวิตช์นับขึ้นลง
  Sec = 0;            // เคลียร์ค่าตัวนับเวลาการกระพริบ
  Dark = 0;          // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segment
}

if(input(PIN_B6) == 0){ //หากมีการกดสวิตช์ยกเลิก CANCEL SW_PIN_B6
  ReadData_EEPROM(); // เรียกโปรแกรมย่อยอ่านหน่วยความจำจากอีพรอม
  puse = 0;           // เคลียร์ค่าตัวแปรตรวจสอบการกดสวิตช์นับขึ้นลง
  Sec = 0;            // เคลียร์ค่าตัวนับเวลาการกระพริบ
  Dark = 0;          // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segment
}
}
}
//////////โปรแกรมนับขึ้น//////////
if(input(PIN_B4)==0 && times < 300){ //ถ้ากดสวิตช์นับขึ้นและ times < 300
  times = (times+1); // เพิ่มค่าตัวแปร times
  step = 1;          // ให้ step = 1 คือนับทีละสเต็ป
  Dark = 0;          // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segmentในขณะกดสวิตช์
}

if(input(PIN_B4)==0 && times == 300){ //ถ้ากดสวิตช์นับขึ้นและ times = 300(กดค้างเกิน3วินาที)
  step = 0;          // ให้ step = 0 คือนับแบบเร็ว
  chk1 = 0;          // เคลียร์ค่าตัวแปรchk1 เป็นตัวตรวจสอบการปล่อยสวิตช์
  Dark = 0;          // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segmentในขณะกดสวิตช์
}

if(input(PIN_B4)==0 && chk1==0){ //ถ้ากดสวิตช์นับขึ้นและค่าตัวแปรchk1=0
  if(counter < 1079){ // ถ้าตัวแปร counter < 1079
    delay_ms(100); //เรียกฟังก์ชันหน่วงเวลา100 ms
    counter = (counter+1); // เพิ่มค่าตัวแปร counter
    chk1 = step; // ตรวจสอบโหมดการนับจากตัวแปร step
  }
}

if(input(PIN_B4)==1){ //ถ้าสวิตช์ PIN_B4ถูกปล่อยแล้ว
  chk1=0; // เคลียร์ค่าตัวแปรchk1 เป็นตัวตรวจสอบการปล่อยสวิตช์
  times=0; // เคลียร์ค่าตัวแปร times
  step=0; // เคลียร์ค่าตัวแปร step
}
}
//////////โปรแกรมนับลง//////////
if(input(PIN_B5)==0 && times2 < 300){ //ถ้ากดสวิตช์นับลงและ times2 < 300
  times2 = (times2+1); // เพิ่มค่าตัวแปร times2
  step2 = 1; // ให้ step2 = 1 คือนับทีละสเต็ป
  Dark = 0; // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segmentในขณะกดสวิตช์
}

if(input(PIN_B5)==0 && times2 == 300){ //ถ้ากดสวิตช์นับลงและ times2 = 300(กดค้างเกิน3วินาที)
  step2 = 0; // ให้ step2 = 0 คือนับแบบเร็ว
  chk2 = 0; // เคลียร์ค่าตัวแปรchk2 เป็นตัวตรวจสอบการปล่อยสวิตช์
  Dark = 0; // เคลียร์ค่าตัวแปรควบคุมการดับของ7 Segmentในขณะกดสวิตช์
}

if(input(PIN_B5)==0 && chk2==0){ //ถ้ากดสวิตช์นับขึ้นและค่าตัวแปรchk2=0
  if(counter > 880){ // ถ้าตัวแปร counter>880
    delay_ms(100); //เรียกฟังก์ชันหน่วงเวลา100 ms
    counter = (counter-1); // ลดค่าตัวแปร counter
    chk2 = step2; // ตรวจสอบโหมดการนับจากตัวแปร step2
  }
}

if(input(PIN_B5)==1){ //ถ้าสวิตช์ PIN_B5ถูกปล่อยแล้ว
  chk2=0; // เคลียร์ค่าตัวแปรchk2 เป็นตัวตรวจสอบการปล่อยสวิตช์
  times2=0; // เคลียร์ค่าตัวแปร times2
  step2=0; // เคลียร์ค่าตัวแปร step
}
}
}
//////////โปรแกรมย่อยแสดงผล//////////

```

```

void Display(void);
void Display(void)
{
switch(scan){
case 0:
//สแกนในหลักที่1
output_low(PIN_A3); //ให้ PIN_A3 เป็นลอจิก"0"คือก่อนนี้เป็น"1"
Disp = num_led[Display_Buff[0]]; //ให้ Display_Buff[0] ซีตำแหน่ง มาเก็บใน Disp
if (Display_Buff[0] == 0){ //ถ้าขอมูลใน Display_Buff[0] = 0
Disp = 0; // สิ่งไม่ให้คอลัมน์1นี้แสดงผล
}
if(Dot == 1){ //ถ้า Dot = 1
Disp = Disp|0x80; // ออร์ค่า Disp ด้วย0x80 เป็นการทำให้ Dot ในหลักที่1ติด
}
if(Dark==1){ //ถ้า Dark=1
Disp = 0; // เช็ดไม่ให้คอลัมน์1นี้แสดงผล
}
output_c(Disp); // ส่งข้อมูลออกพอร์ต C
output_high(PIN_A0); // ส่งให้คอลัมน์ที่1 ติด
scan=1; //ส่งค่าเพื่อให้สแกนในหลักที่2
break; //สิ้นสุดการสแกนหลักที่1
case 1:
//สแกนในหลักที่2
output_low(PIN_A0); //ให้ PIN_A0 เป็นลอจิก"0"คือก่อนนี้เป็น"1"
Disp = num_led[Display_Buff[1]]; //ให้ Display_Buff[1] ซีตำแหน่ง มาเก็บใน Disp
if (Display_Buff[1] == 0 && Display_Buff[0] == 0){ //ถ้าขอมูลใน Display_Buff[0,1] = 0
Disp = 0; // สิ่งไม่ให้คอลัมน์2นี้แสดงผล
}
if(Dot == 2){ //ถ้า Dot = 2
Disp = Disp|0x80; // ออร์ค่า Disp ด้วย0x80 เป็นการทำให้ Dot ในหลักที่2ติด
}
if(Dark==1){ //ถ้า Dark=1
Disp = 0; // เช็ดไม่ให้คอลัมน์2นี้แสดงผล
}
output_c(Disp); // ส่งข้อมูลออกพอร์ต C
output_high(PIN_A1); // ส่งให้คอลัมน์ที่2 ติด
scan = 2; //ส่งค่าเพื่อให้สแกนในหลักที่3
break; //สิ้นสุดการสแกนหลักที่2
case 2:
//สแกนในหลักที่3
output_low(PIN_A1); //ให้ PIN_A1 เป็นลอจิก"0"คือก่อนนี้เป็น"1"
Disp = num_led[Display_Buff[2]]; //ให้ Display_Buff[2] ซีตำแหน่ง มาเก็บใน Disp
if (Display_Buff[2] == 0 && Display_Buff[1] == 0
&& Display_Buff[0] == 0){ //ถ้าขอมูลใน Display_Buff[0,1,2] = 0
Disp = 0; // สิ่งไม่ให้คอลัมน์3นี้แสดงผล
}
if(Dot == 3){ //ถ้า Dot = 3
Disp = Disp|0x80; // ออร์ค่า Disp ด้วย0x80 เป็นการทำให้ Dot ในหลักที่3ติด
}
if(Dark==1){ //ถ้า Dark=1
Disp = 0; // เช็ดไม่ให้คอลัมน์3นี้แสดงผล
}
output_c(Disp); // ส่งข้อมูลออกพอร์ต C
output_high(PIN_A2); // ส่งให้คอลัมน์ที่3 ติด
scan=3; //ส่งค่าเพื่อให้สแกนในหลักที่4
break; //สิ้นสุดการสแกนหลักที่3
case 3:
//สแกนในหลักที่4
output_low(PIN_A2); //ให้ PIN_A2 เป็นลอจิก"0"คือก่อนนี้เป็น"1"
Disp = num_led[Display_Buff[3]]; //ให้ Display_Buff[3] ซีตำแหน่ง data มาเก็บใน Disp
if(Dot == 4){ //ถ้า Dot = 4
Disp = Disp|0x80; // ออร์ค่า Disp ด้วย0x80 เป็นการทำให้ Dot ในหลักที่4ติด
}
if(Dark==1){ //ถ้า Dark=1
Disp = 0; // เช็ดไม่ให้คอลัมน์4นี้แสดงผล
}
output_c(Disp); // ส่งข้อมูลออกพอร์ต C
output_high(PIN_A3); // ส่งให้คอลัมน์ที่4 ติด
scan = 0; //ส่งค่าเพื่อให้สแกนในหลักที่1
break; //สิ้นสุดการสแกนหลักที่4
}
}
//////////////////////////////////////โปรแกรมย่อยอินเตอร์ปต์//////////////////////////////////////
#INT_TIMER1 //ประกาศใช้อินเตอร์ปต์ใหม่เมอร์1
void IntTMR1_isr(void){

```

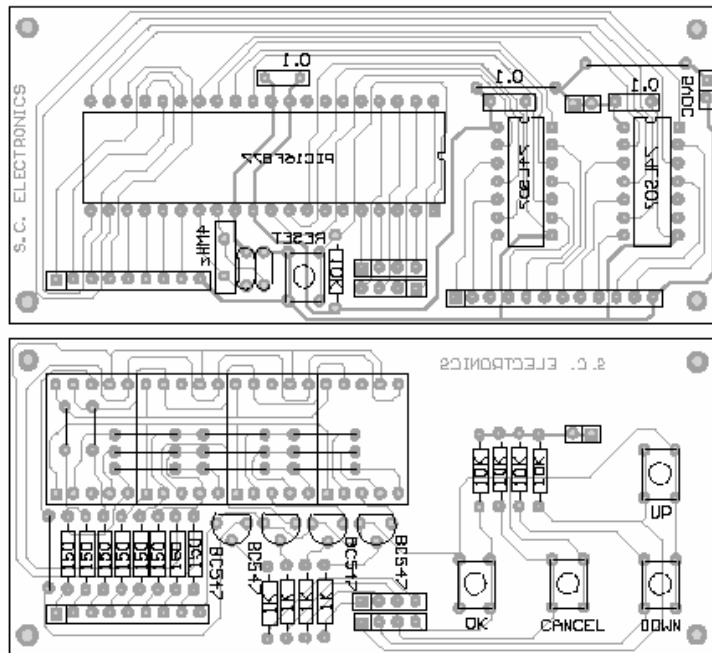
```

Display(); //เรียกโปรแกรมย่อยแสดงผล
set_timer1(65000); //ตั้งค่าเริ่มต้นของการนับไทม์เมอร์1ที่ 65000
}
//////////////////////////////////////////////////โปรแกรมหลัก//////////////////////////////////////
void main(void);
void main(void)
{
enable_interrupts(GLOBAL); //เปิดใช้งานอินเตอร์รัปต์
enable_interrupts(INT_TIMER1); //เปิดใช้อินเตอร์รัปต์ไทม์เมอร์1
setup_timer_1(T1_INTERNAL | T1_DIV_BY_1); //ตั้งค่าปริสเกลเลอร์ในที่นี้คือหาร1
set_timer1(65000); //ตั้งค่าเริ่มต้นของการนับไทม์เมอร์1ที่ 65000
while(TRUE) //ลูป
{
if(r == 1){ReadData_EEPROM();} //ถ้า r = 1เป็นการอ่านข้อมูลจาก EEPROM ในครั้งแรก
if(counter<880||counter>1079){ counter = 880;} //หากข้อมูลที่อ่านมาไม่อยู่ในช่วง 880 ถึง 1079ให้ //counter = 880
if(w ==1){WriteData_EEPROM();} //ถ้า r = 1เขียนข้อมูลลง EEPROM
Input_sw(); //เรียกโปรแกรมย่อยรับคำสั่งรีเซ็ต
Converse(); //เรียกโปรแกรมย่อยแยกค่าตัวเลขประจำหลัก
}
}

```

การสร้าง

ในเครื่องต้นแบบจะแยกส่วนของแหล่งจ่ายไฟออกมาอยู่ข้างนอกเป็นชุดกล่องเพาเวอร์ซัพพลาย เพื่อให้เครื่องไม่เกิดความร้อน และสัญญาณรบกวนจากหม้อแปลง ไปทำให้การทำงานของวงจรรอสซิลิเตอร์และวงจรรีเลย์ผิดพลาดได้ แผ่นวงจรพิมพ์จะแบ่งออกเป็นสองแผ่นดังรูปที่2 โดยทั้งสองแผ่นจะใช้ขั้วต่อซึ่งมีตำแหน่งตรงกันพอดียึดแผ่นวงจรทั้งสองและใช้น็อตขัน โดยใช้ บขพลาสติกกรองตั้งรูป หรืออาจใช้วิธีอื่นก็ได้ตามแต่สะดวก ในการเฟลชโปรแกรมชิพนั้น เครื่องต้นแบบใช้โปรแกรม Miracle PIC สามารถหาดาวน์โหลดได้ฟรี ส่วนบอร์ดเฟลชโปรแกรมหากท่านใดมีแล้วักใช้ได้ทันทีแต่ถ้ายังไม่มียังสามารถสร้างเองได้ไม่ยากรายละเอียดการสร้างบอร์ดสำหรับเฟลชโปรแกรมจะกล่าวในตอนท้าย



รูปที่2 แผ่นวงจรพิมพ์

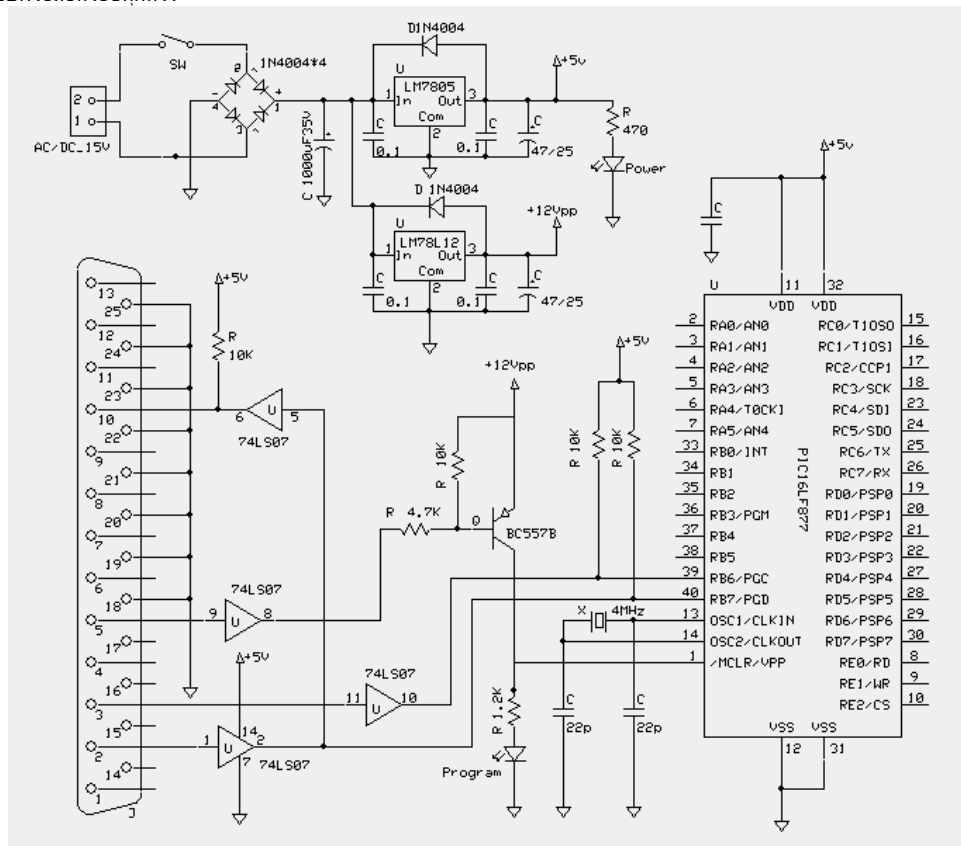
ก่อนอื่นต้องได้แผ่นวงจรพิมพ์มาก่อนจะออกแบบเองหรือใช้ตามเครื่องต้นแบบก็ได้เมื่อได้แผ่นวงจรพิมพ์แล้วลงอุปกรณ์โดยเริ่มจากจัมเปอร์ก่อนตามด้วยตัวต้านทานระง้อขั้วให้มีการช้อดกันของจุดบัดกรีไอซีทุกตัวควารใส่ช้อกเก็ด เนื่องจากลายปรินท์มีขนาดเล็กมากเมื่อลงอุปกรณ์ครบแล้วอย่าเพิ่งประกบบอร์ดทั้งสองเข้าด้วยกันให้ใช้มัลติมิเตอร์ตั้งย่าน R*1 วัด 7Segment ว่ามีการขาดหรือช้อดของลายปรินท์หรือไม่โดยวัดจากตำแหน่งขั้วเสียบที่ต่อกับพอร์ตกับคอมมอนของแต่ละหลักหรือขาคอลเล็กเตอร์ของทรานซิสเตอร์ BC547 แต่ละตัวจรจุดบัดกรีทุกจุดให้ดีหากไม่มั่นใจใช้มิเตอร์ช่วย



รูปที่3 เครื่องต้นแบบที่ประกอบเสร็จแล้ว

สร้างบอร์ดสำหรับแฟลชโปรแกรม PIC16F877

สำหรับท่านที่ยังไม่มีบอร์ดแฟลชโปรแกรม วงจรรูปที่4เป็นวงจรสำหรับแฟลชโปรแกรม PIC16F877 ที่ใช้ได้ผลดีเหมือนกันโดยใช้ร่วมกับซอฟต์แวร์ Miracle PIC สามารถใช้ได้ทั้งไฟตรงและสลับ 14 -18 โวลต์ไม่แนะนำให้ใช้แรงดันสูงหรือต่ำกว่านี้เพราะแรงดันสูงเกินไปไอซีเรีกกulet5Vจะร้อน ส่วนถ้าแรงดันต่ำเกินไปไอซีเรีกกuletแรงดัน 12V สำหรับโปรแกรมจะทำการรักษาแรงดันให้คงที่ไม่ได้ ดีเท่าที่ควร คอนเน็กเตอร์ที่ใช้คือคอนเน็กเตอร์ พอร์ต ขนาน หรือ DB25 ไม่ควรใช้สายยาวเกินความจำเป็นเพราะสัญญาณที่พอร์ตขนานเป็นแรงดันต่ำเกิดการสูญเสียได้ง่าย วิธีการสร้างในเครื่องต้นแบบใช้วิธี ไวร่สายบนแผ่นปรินท์อเนกประสงค์ เพราะวงจรไม่ซับซ้อนมากนักหรือใครจะออกแบบลายปรินท์ก็ได้ ข้อควรระวังในการใช้งาน อย่าลืมปิดสวิตซ์หากมีการเสียบหรือถอดไอซีทุกครั้ง



รูปที่ 4 วงจรที่ใช้สำหรับสำหรับโปรแกรม PIC16F877