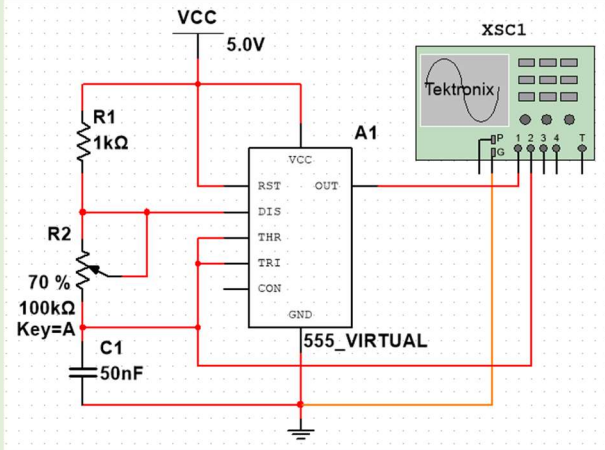
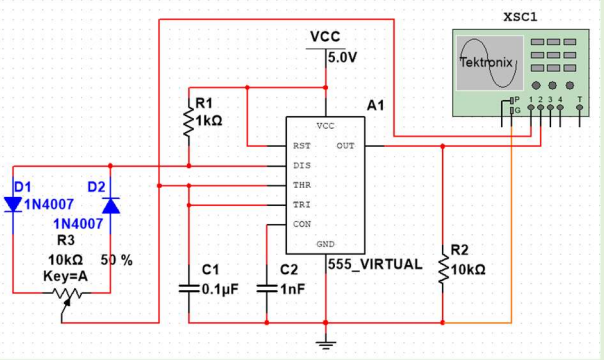
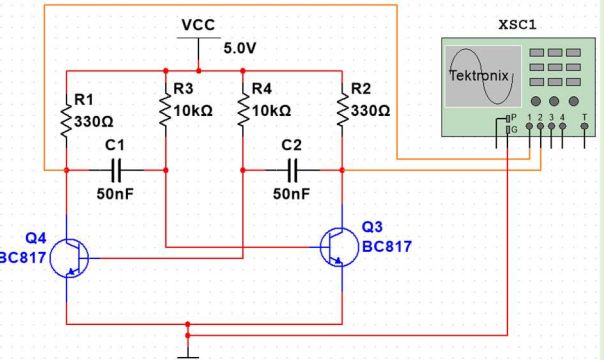
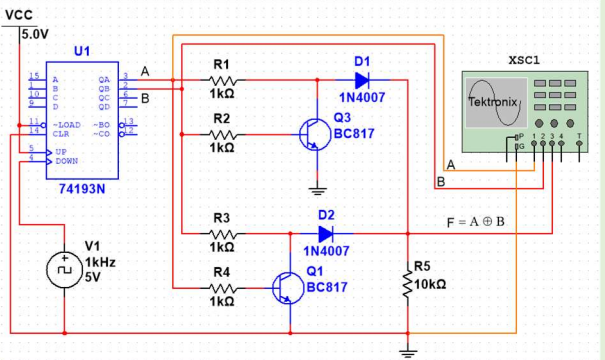
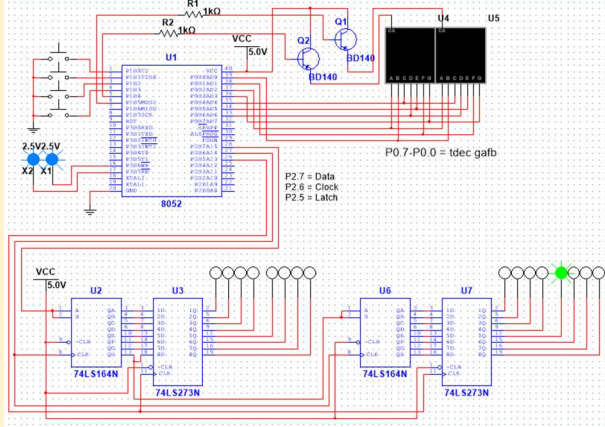
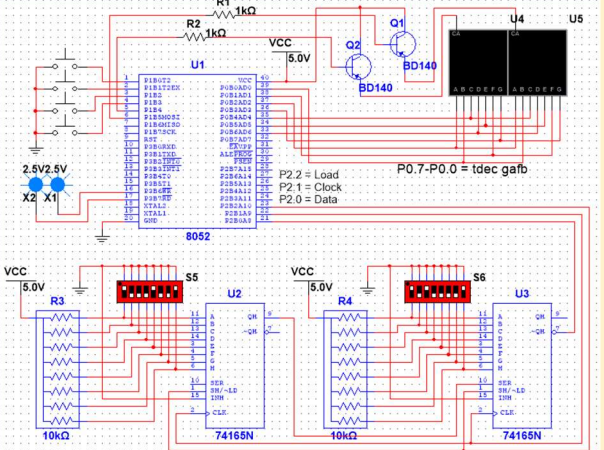



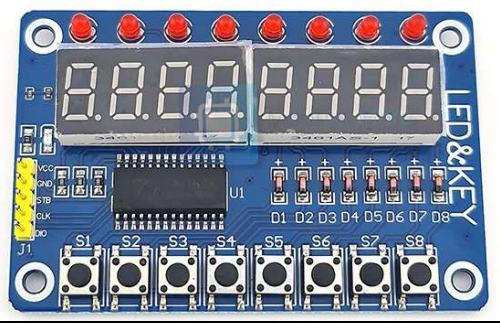


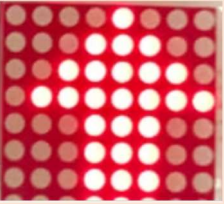
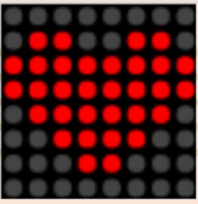
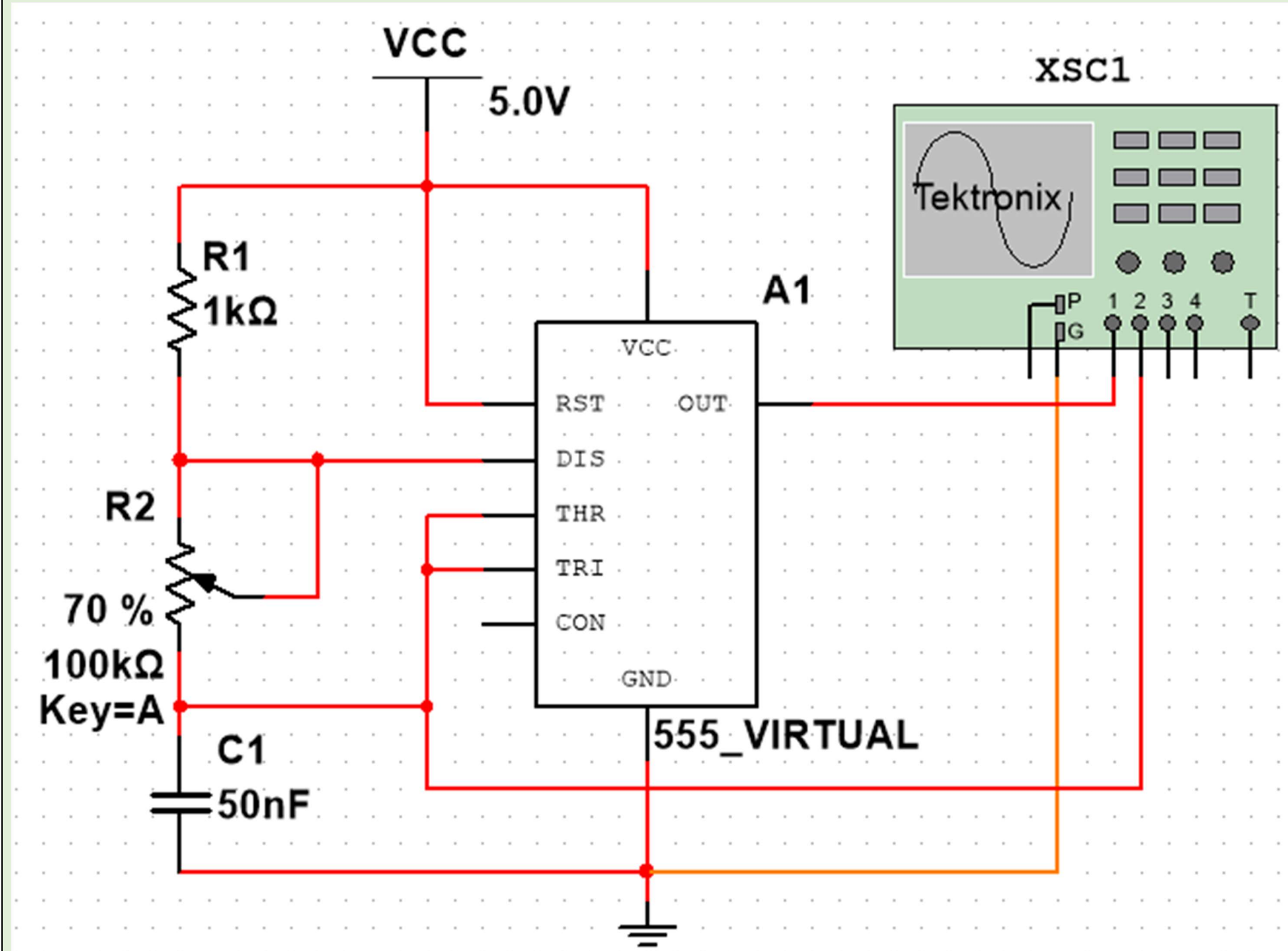


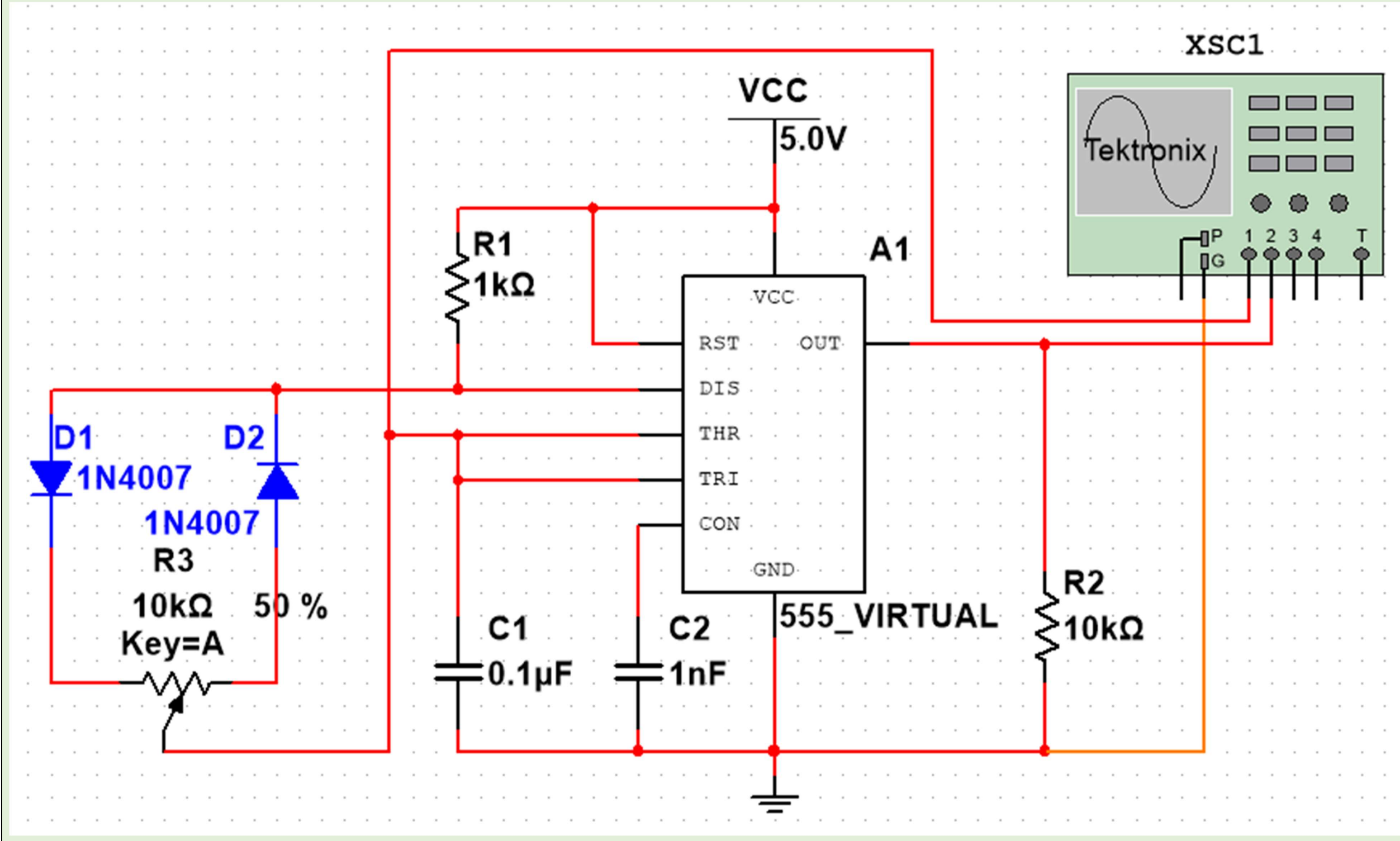
	A	B	C	D
1. Circuit	IC 555 -Timer	IC 555-PWM	BJT Astable Multivibrator	F = A⊕B Timing
				
Step 11 = 5 คะแนน	จำลองการทำงานใน MutiSim14	จำลองการทำงานใน MutiSim14	จำลองการทำงานใน MutiSim14	จำลองการทำงานใน MutiSim14
Step 12 = +5 คะแนน	ต้องวงจร แล้ววัดสัญญาณให้ได้ตามการจำลองการทำงาน	ต้องวงจร แล้ววัดสัญญาณให้ได้ตามการจำลองการทำงาน	ต้องวงจร แล้ววัดสัญญาณให้ได้ตามการจำลองการทำงาน	ต้องวงจร แล้ววัดสัญญาณให้ได้ตามการจำลองการทำงาน
	https://778139821211757900.weebly.com/1kh-z-clock.html	https://www.circuits-diy.com/dc-motor-speed-controlling-circuit-pwm/#google_vignette	https://bestengineeringprojects.com/clock-signal-generator-circuit/	https://www.edn.com/simple-xor-logic-elements-on-transistors/
		https://www.build-electronic-circuits.com/555-pwm-circuit/	https://www.electronics-tutorials.ws/waveforms/astable.html	
2. Circuit + Code	Serial Shift Output 16 Bit with Display	Serial Shift Input 16 Bit with Display		
				
Step 21 = 5 คะแนน	จำลองการทำงานใน MutiSim14	จำลองการทำงานใน MutiSim14		
Step 22 = +5 คะแนน	ปรับวงจรและโปรแกรมให้เหลือ 8 บิต ต้องวงจร โหลดโค้ดให้ทำงาน	ปรับวงจรและโปรแกรมให้เหลือ 8 บิต ต้องวงจร โหลดโค้ดให้ทำงาน		
Step 23 = +5 คะแนน	ต้องวงจร โหลดโค้ดให้ทำงานแบบ 16บิต ตามการจำลองการทำงาน	ต้องวงจร โหลดโค้ดให้ทำงานแบบ 16บิต ตามการจำลองการทำงาน		

3. ESP32 + IO	ESP32 + 7_Segment 1Digit Circuit	ESP32 + 7_Segment 4Digit Circuit		
Mission 31 +4 คะแนน	Display 0-F ติดทีละตัว 	Display 0-F ติดทีละตัวตำแหน่งซ้ายสุด 	X	X
Mission 32 +4 คะแนน	Display your ID เช่น b6733377 b -> ดับ -> 6 -> ดับ -> 7 -> ดับ -> 3 -> ดับ -> 3 -> ดับ -> 3 -> ดับ -> 7 -> ดับ -> 7 -> ดับ ->.... (วนรอบไปเรื่อย ๆ ต่อเนื่อง ตลอดเวลา)	Display your ID เช่น b6733377 ---- -> ---b -> --b6 -> -b67 -> b673 -> 6733 -> 7333 -> ... (ไหลไปทางซ้ายทีละตัว ต่อเนื่อง ตลอดเวลา)		
4. ESP32 + IO	ESP32 Circuit 4A	ESP32 Circuit 4B		
Mission 41 +4 คะแนน	MAX-7219 แสดงรหัสของตัวเอง 	TM-1638 แสดงรหัสของตัวเอง 	X	X
Mission 42 +4 คะแนน	AHT20 อ่านอุณหภูมิ แสดงที่ Serial Terminal 	DHT22 อ่านอุณหภูมิ แสดงที่ Serial Terminal 		
Mission 43 +4 คะแนน	8x8 Dot Matrix LED แสดงลูกศรลิฟท์กำลังขึ้น  https://www.hackster.io/LisleapexBlog/max7219-arduino-program-8x8-cathode-led-dot-matrix-display-ccc895	8x8 Dot Matrix LED แสดงรูปหัวใจ  https://iot-kmutnb.github.io/blogs/displays/max7219/		

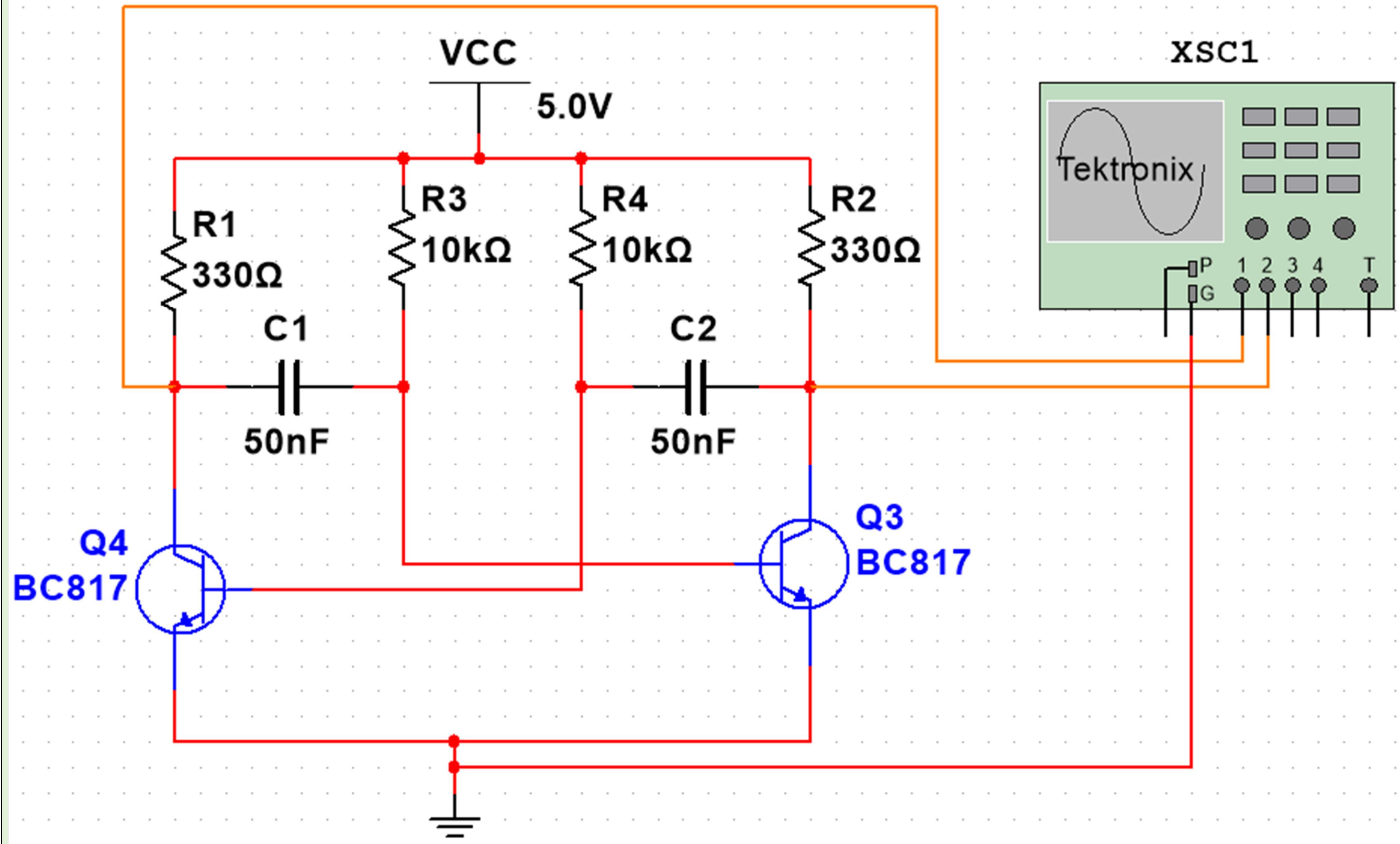
1A: IC 555 -Timer



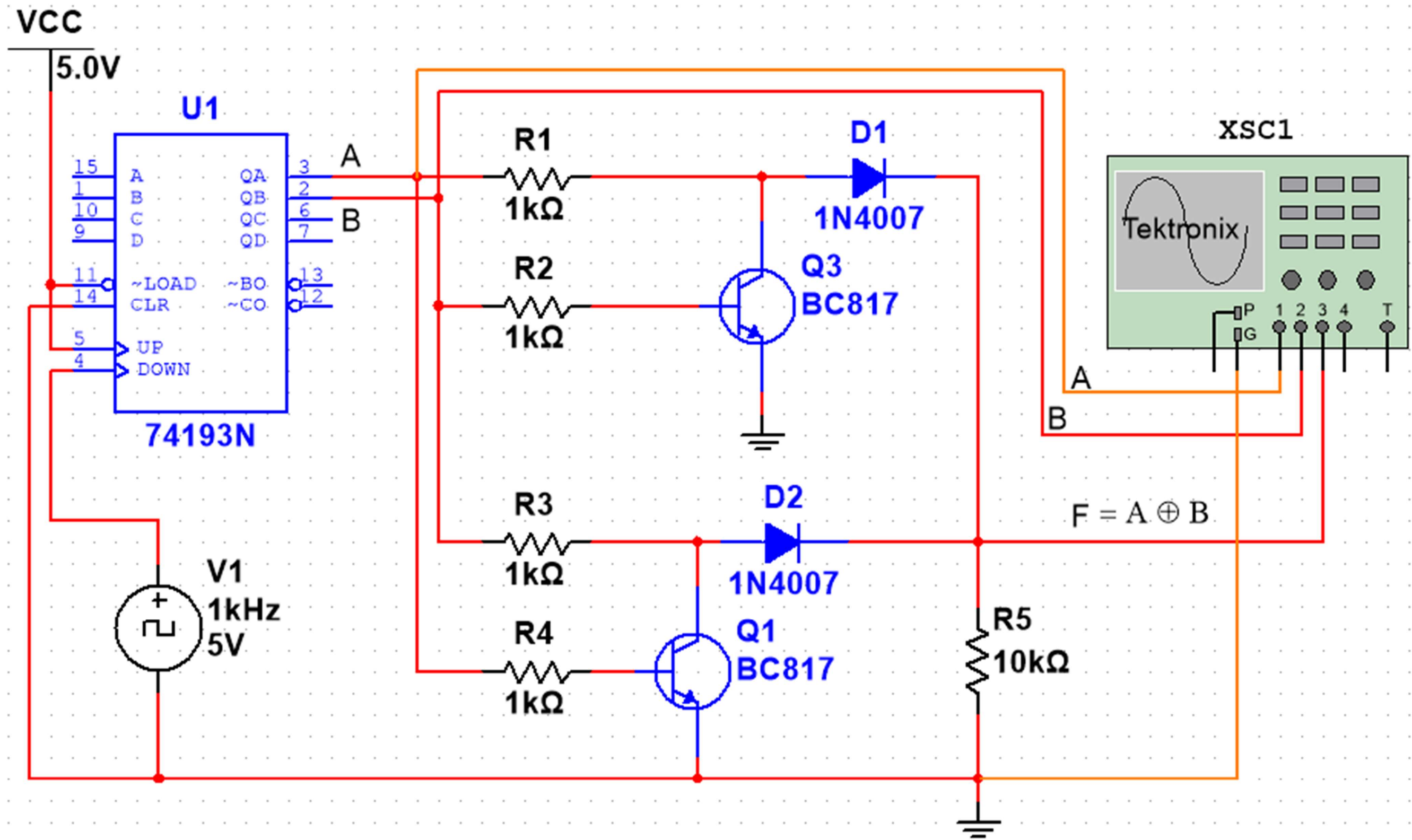
1B: IC 555-PWM



1C: BJT Astable Multivibrator



1D: $F = A \oplus B$ Timming



2A: Serial Shift Output 16 Bit with Display

```
kDelay EQU 1
; if Run on board >> kDelay EQU 25
; if Simulation >> kDelay EQU 1
```

```
LF_SEG EQU P1.4
RG_SEG EQU P1.5
PT_SEG EQU P0
```

```
pData EQU P2.7
pClock EQU P2.6
pLatch EQU P2.5
```

```
ORG 0000H
JMP 0100H
```

```
ORG 0100H
MOV SP,#2FH
CLR LF_SEG
CLR RG_SEG
```

```
LOOP: MOV DPTR,#0001H
CALL Send16_LED
MOV DPTR,#0002H
CALL Send16_LED
MOV DPTR,#0004H
CALL Send16_LED
MOV DPTR,#0008H
CALL Send16_LED
MOV DPTR,#0010H
CALL Send16_LED
MOV DPTR,#0020H
CALL Send16_LED
MOV DPTR,#0040H
CALL Send16_LED
MOV DPTR,#0080H
CALL Send16_LED
MOV DPTR,#0100H
CALL Send16_LED
MOV DPTR,#0200H
CALL Send16_LED
MOV DPTR,#0400H
CALL Send16_LED
MOV DPTR,#0800H
CALL Send16_LED
MOV DPTR,#1000H
CALL Send16_LED
MOV DPTR,#2000H
CALL Send16_LED
MOV DPTR,#4000H
CALL Send16_LED
MOV DPTR,#8000H
CALL Send16_LED
JMP LOOP
```

```
Send16_LED:
MOV A,DPL
CALL Shift8
MOV A,DPH
CALL Shift8
CLR pLatch
SETB pLatch
CALL Display_DPTR
MOV R7,#10
_Wait0: MOV R6,#kDelay
_Wait1: CALL DELAY
DJNZ R6,_Wait1
DJNZ R7,_Wait0
RET
```

```
Shift8: MOV B,#8
_Shift: RRC A
MOV pData,C
CLR pClock
SETB pClock
DJNZ B,_Shift
RET
```

```
Display_DPTR:
MOV A,#10H
CALL _Send1Seg
MOV A,DPH
CALL _Send2HEX
MOV A,DPL
CALL _Send2HEX
RET
```

```
_Send2HEX:
PUSH Acc
SWAP A
CALL _Send1HEX
POP Acc
```

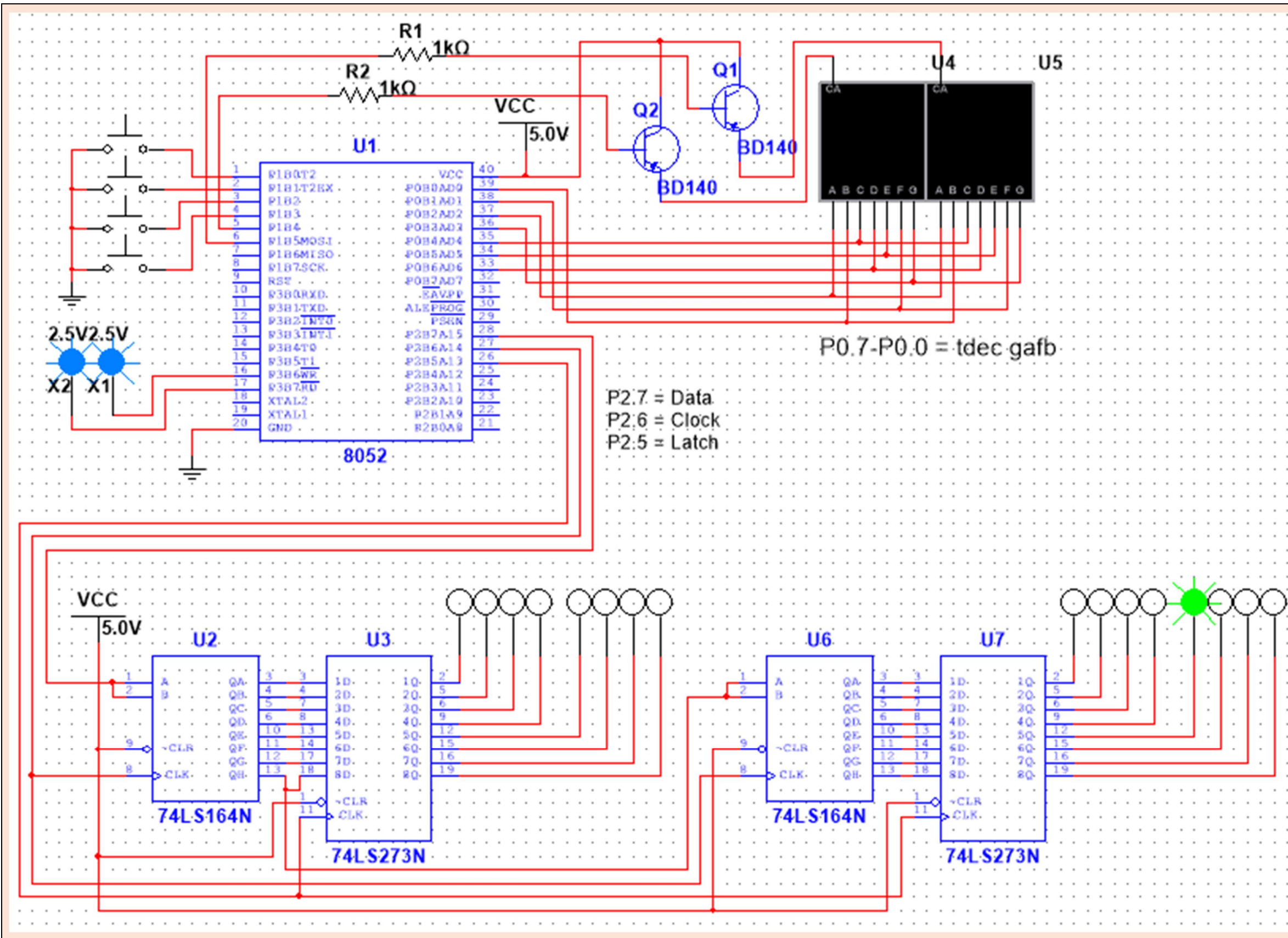
```
_Send1HEX:
ANL A,#0FH
```

```
_Send1Seg:
PUSH DPH
PUSH DPL
MOV DPTR,#T_SEG
MOVC A,@A+DPTR
CPL A
MOV PT_SEG,A
POP DPL
POP DPH
MOV R7,#kDelay
```

```
_Wait2: CALL DELAY
DJNZ R7,_Wait2
MOV PT_SEG,#0FFH
MOV R7,#kDelay
_Wait3: CALL DELAY
DJNZ R7,_Wait3
RET
```

```
T_SEG: DB 77H, 11H, 6DH, 5DH, 1BH, 5EH, 7EH, 15H ; Code 0123 4567
DB 7FH, 5FH, 3FH, 7AH, 66H, 79H, 6EH, 2EH ; Code 89ab cdef
DB 08H
```

```
Delay: MOV B,#kDelay
MOV A,#250
_DLY00: DJNZ Acc,_DLY00
DJNZ B,_DLY00
RET
END
```



2B: Serial Shift Input 16 Bit with Display

```
kDelay EQU 1
; if Run on board >> kDelay EQU 25
; if Simulation >> kDelay EQU 1
```

```
LF_SEG EQU P1.4
RG_SEG EQU P1.5
PT_SEG EQU P0
```

```
pLED_R EQU P3.6
pLED_G EQU P3.7
```

```
pLoad EQU P2.2
pClock EQU P2.1
pData EQU P2.0
Data_H EQU 20H
Data_L EQU 21H
```

```
ORG 0000H
JMP 0100H
```

```
ORG 0100H
CLR EA
MOV SP,#3FH
SETB pData
CLR LF_SEG
CLR RG_SEG
```

```
LOOP: CALL GET_16BitData
CALL Display_DATA
MOV R7,#10
_Wait0: MOV R6,#kDelay
_Wait1: CALL DELAY
DJNZ R6,_Wait1
DJNZ R7,_Wait0
JMP LOOP
```

```
GET_16BitData:
CLR pLoad
SETB pLoad
CALL _Get8B
MOV Data_L,A
CALL _Get8B
MOV Data_H,A
RET
```

```
_Get8B: MOV R0,#8
_LP00: MOV C,pData
RRC A
CLR pClock
SETB pClock
DJNZ R0,_LP00
RET
```

```
Display_DATA:
MOV A,#10H
CALL _Send1Seg
MOV A,Data_H
CALL _Send2HEX
MOV A,Data_L
CALL _Send2HEX
RET
```

```
_Send2HEX:
PUSH Acc
SWAP A
CALL _Send1HEX
POP Acc
```

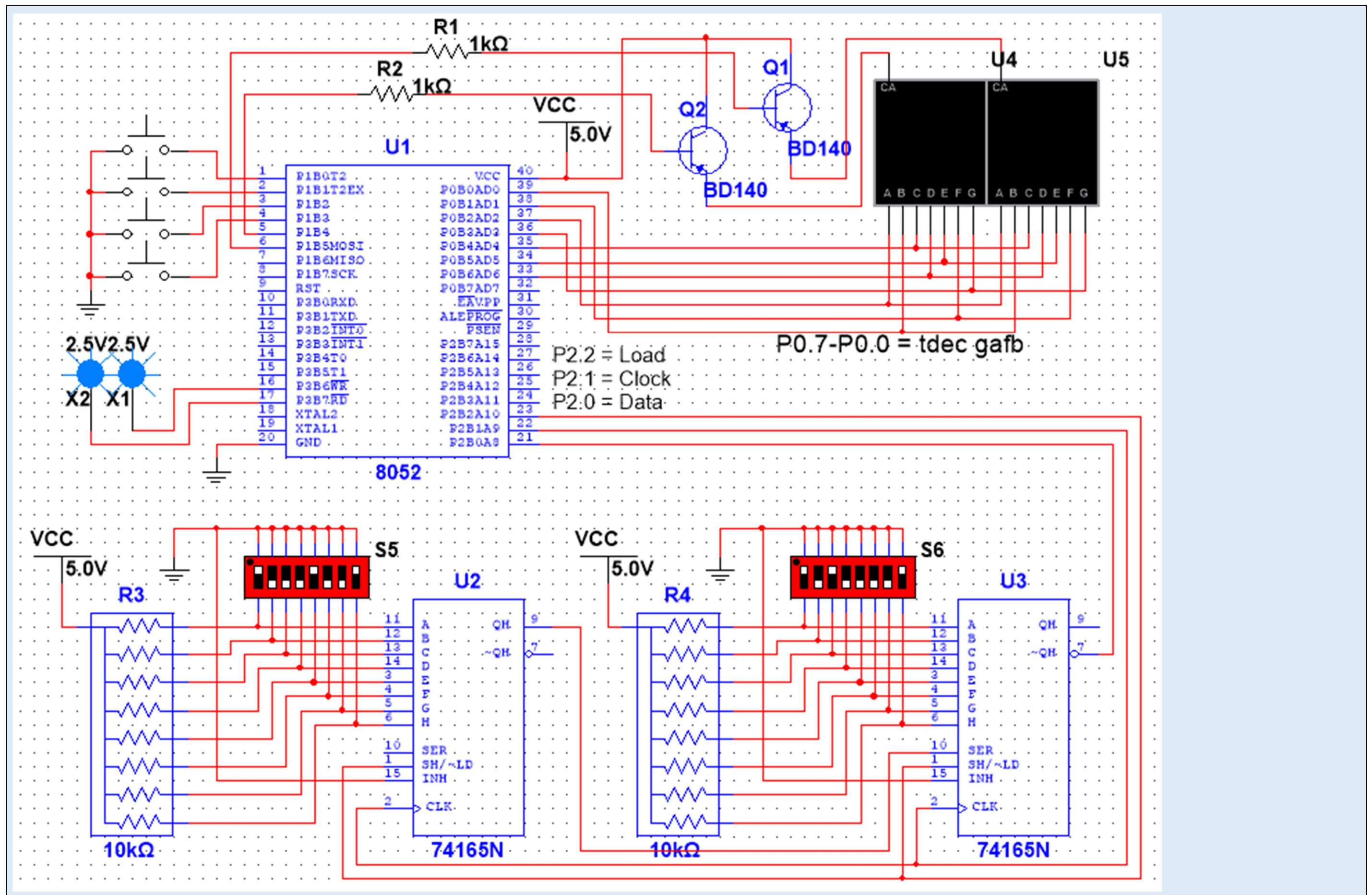
```
_Send1HEX:
ANL A,#0FH
```

```
_Send1Seg:
PUSH DPH
PUSH DPL
MOV DPTR,#T_SEG
MOVC A,@A+DPTR
CPL A
MOV PT_SEG,A
POP DPL
POP DPH
MOV R7,#kDelay
_Wait2: CALL DELAY
DJNZ R7,_Wait2
MOV PT_SEG,#0FFH
MOV R7,#kDelay
_Wait3: CALL DELAY
DJNZ R7,_Wait3
RET
```

```
T_SEG: DB 77H, 11H, 6DH, 5DH, 1BH, 5EH, 7EH, 15H ; Code 0123 4567
DB 7FH, 5FH, 3FH, 7AH, 66H, 79H, 6EH, 2EH ; Code 89ab cdef
DB 08H
```

```
Delay: MOV B,#kDelay
MOV A,#250
_DLY00: DJNZ Acc,_DLY00
DJNZ B,_DLY00
RET
```

```
END
```



กลุ่มเรียน	กลุ่มแลป	กลุ่มสอบ	รหัส	ชื่อสกุล
<input type="checkbox"/> IC 555 -Timer <input type="checkbox"/> IC 555-PWM <input type="checkbox"/> BJT Astable Multivibrator <input type="checkbox"/> $F = A \oplus B$ Timming		<input type="checkbox"/> Serial Shift Output 16 Bit with Display <input type="checkbox"/> Serial Shift Input 16 Bit with Display	7-Segment Circuit <input type="checkbox"/> Display 1 Digit <input type="checkbox"/> Display 4 Digit	ESP32 Interface Circuit <input type="checkbox"/> Display 8 Digit <input type="checkbox"/> Sensor <input type="checkbox"/> 8x8 Dot LED
Simulators: +5 คะแนน		Simulators: +5 คะแนน	Display 0-F +4 คะแนน	Display 8 Digit +4 คะแนน
Circuit: +5 คะแนน		Circuit RUN 8 Bit: +5 คะแนน	Display ID +4 คะแนน	Sensor +4 คะแนน
		Circuit RUN 16 Bit: +5 คะแนน		8x8 Dot LED +4 คะแนน