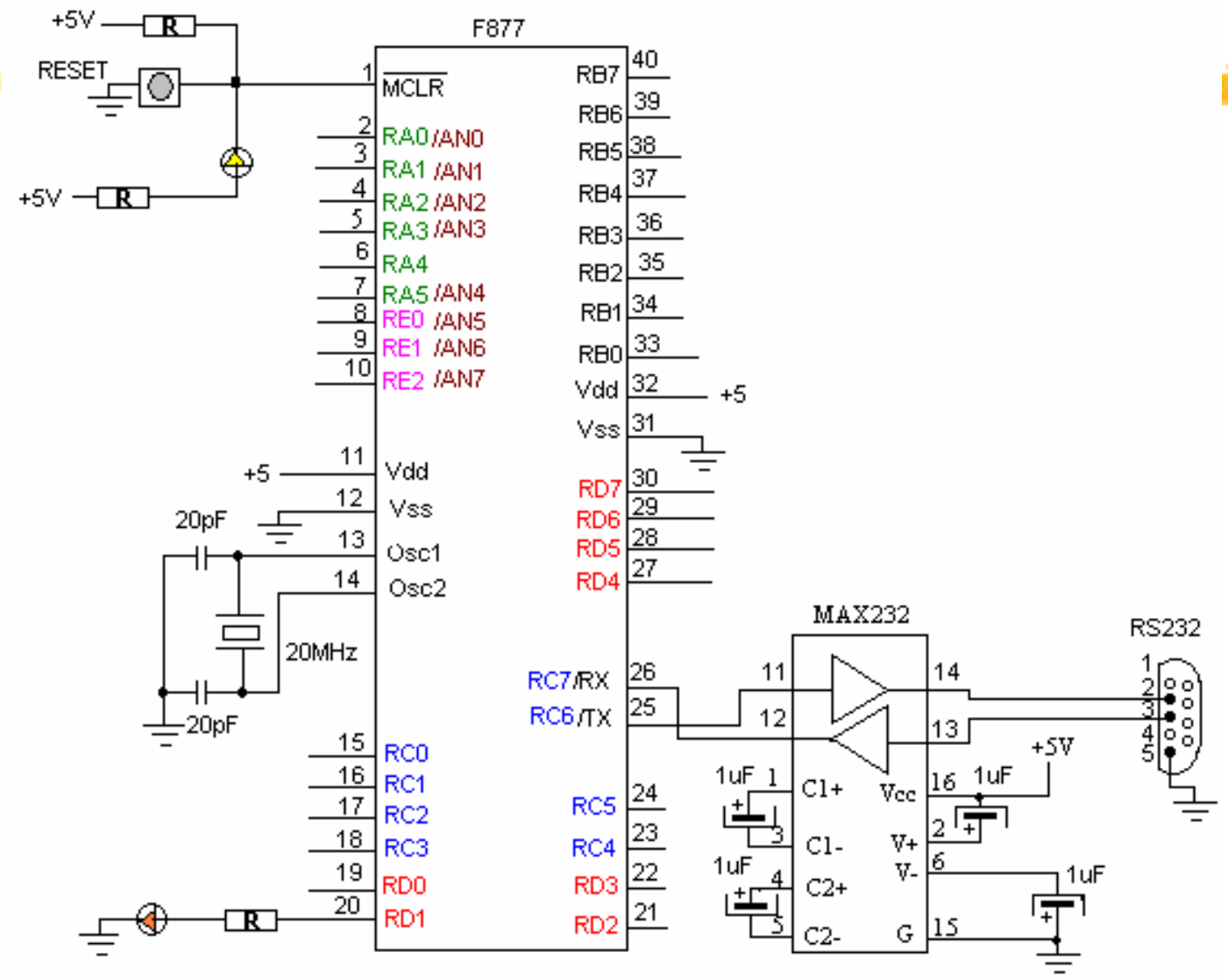
EECE416: Microcomputer Fundamentals and Design

PIC-Coding Practice

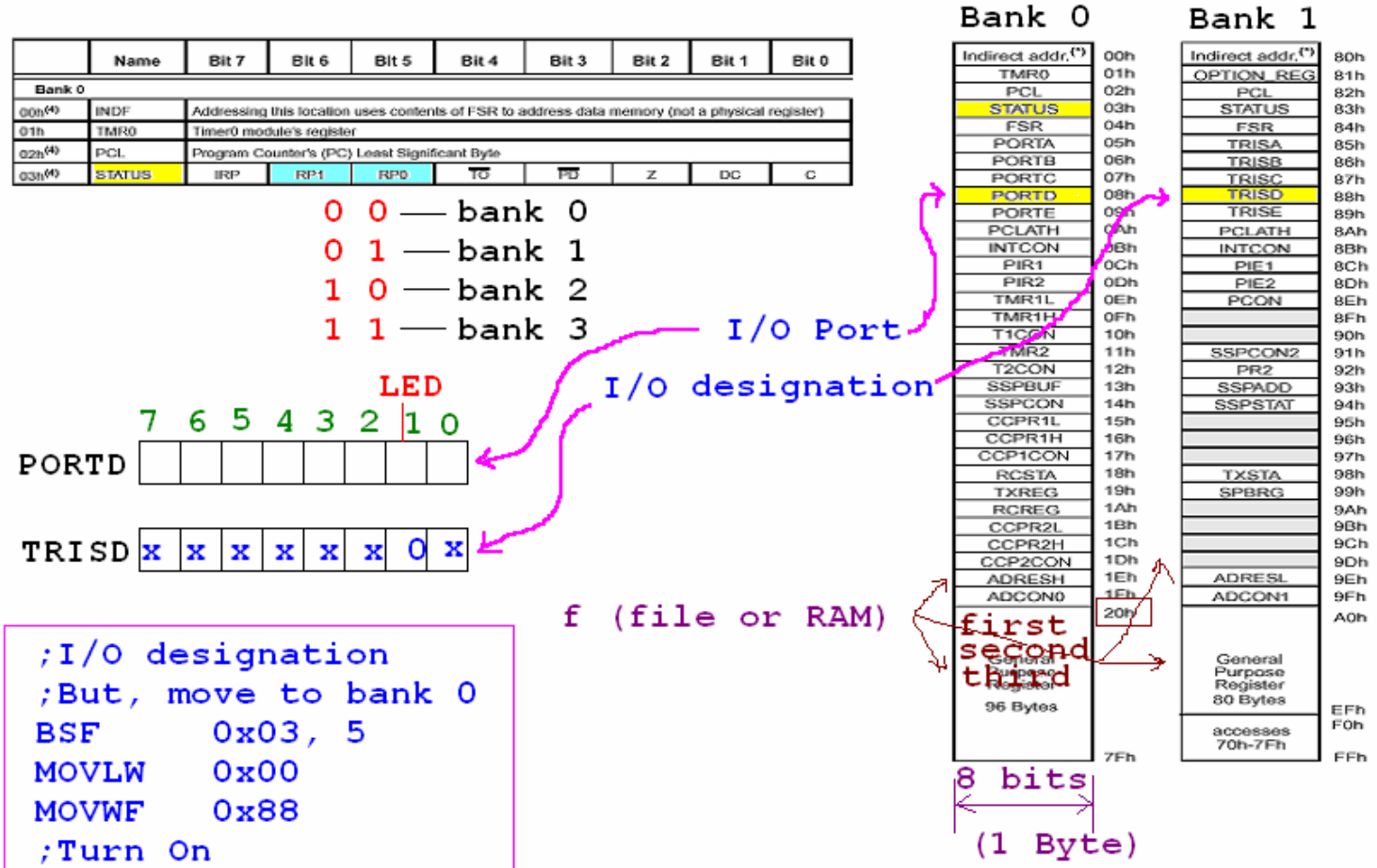
Dr. Charles J. Kim

Howard University

1: LED On/Off

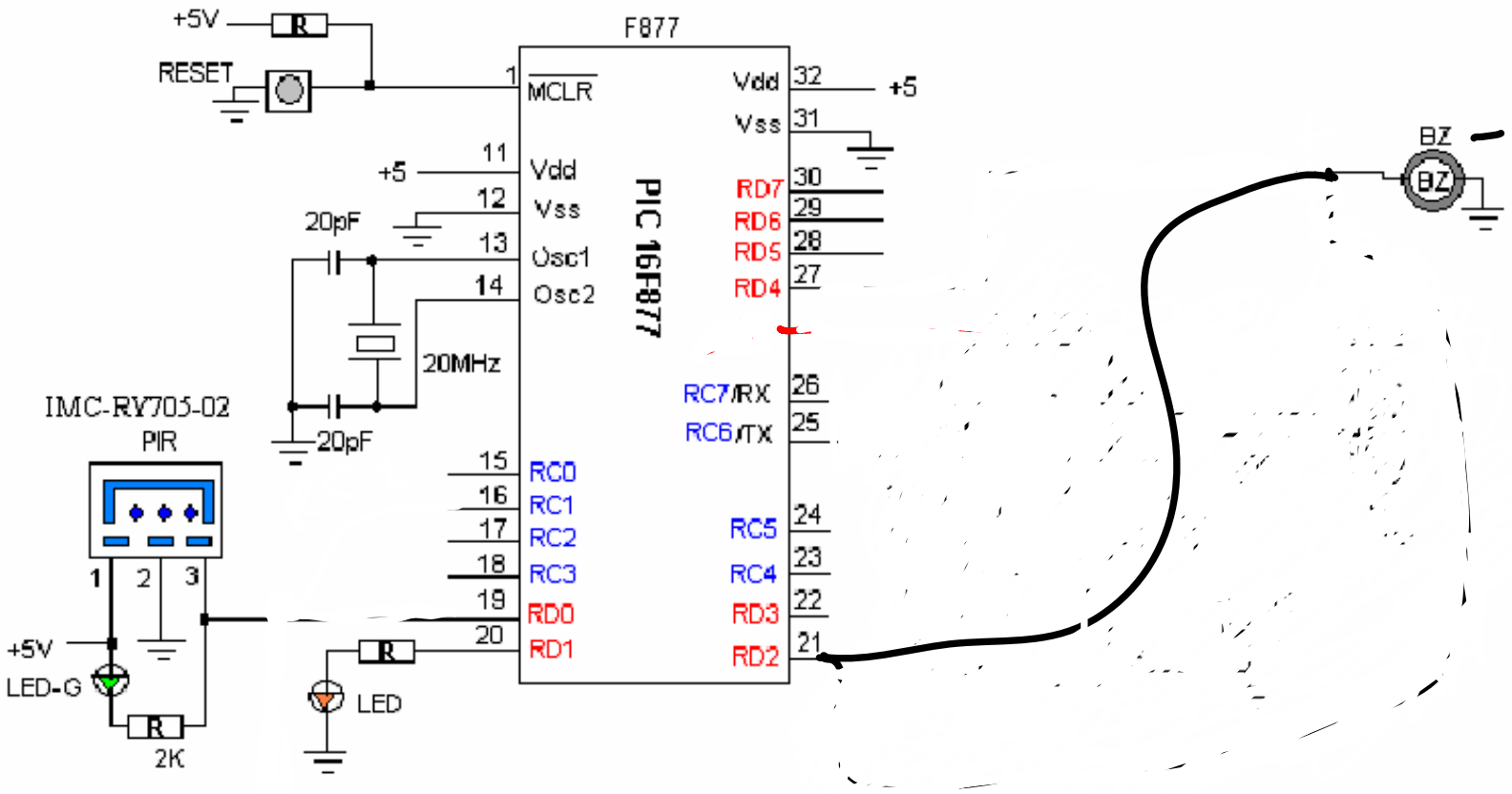

F877
+5V
RESET
+5V
MCLR
RA0 /AN0
RA1 /AN1
RA2 /AN2
RA3 /AN3
RA4
RA5 /AN4
RE0 /AN5
RE1 /AN6
RE2 /AN7
Vdd
Vss
Osc1
Osc2
RC0
RC1
RC2
RC3
RD0
RD1
RB7
RB6
RB5
RB4
RB3
RB2
RB1
RB0
Vdd
Vss
RD7
RD6
RD5
RD4
RC7/RX
RC6/TX
RC5
RC4
RD3
RD2
+5
20pF
20MHz
20pF
MAX232
C1+
C1-
C2+
C2-
Vcc
V+
V-
G
1uF
+5V
RS232

Illustration




```
;I/O designation
;But, move to bank 0
BSF      0x03, 5
MOVLW    0x00
MOVWF    0x88
;Turn On
BSF      0x08, 1
CALL     DELAY
;Turn Off
 BSF     0x08, 1
```

2: Motion Detection and Buzzing



Piezo Buzzer

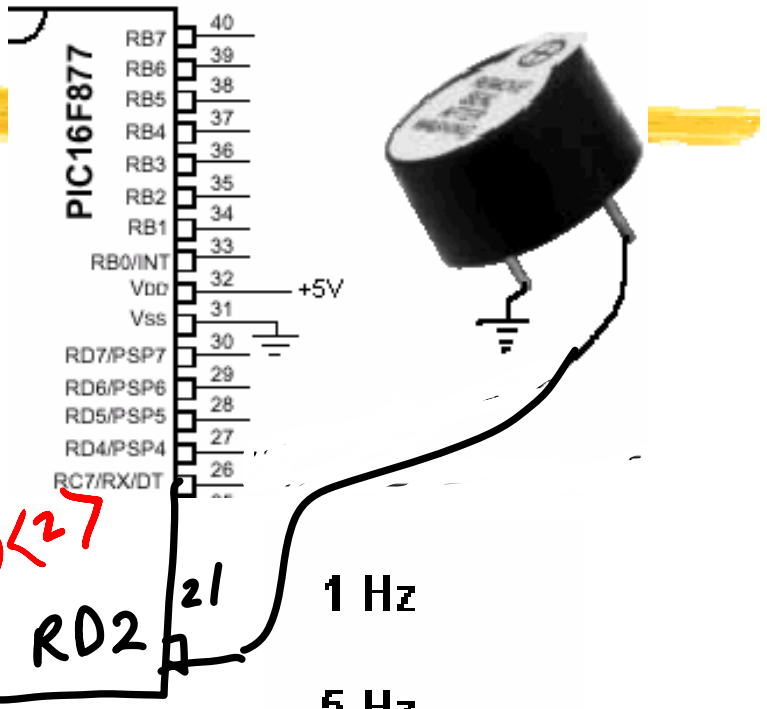
CE-328 Series

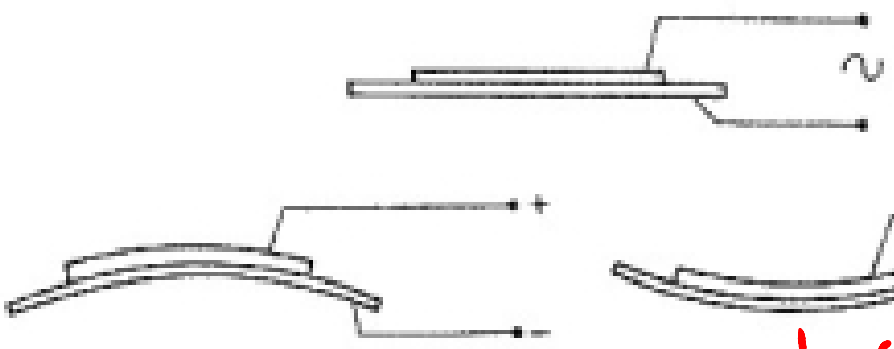
color Black
Noryl
Housing Material
Pin Terminal

ELECTRICAL SPECIFICATIONS

MODEL NO.		32S4120
Operating Voltage	(VDC)	3 - 16
Rated Voltage	(VDC)	12
*Max. Rated Current	(mA.)	7
*Min. Sound Output	(dBA/10cm)	80
*Frequency	(Hz.)	4000±500
Tone Nature		single
Operating Temperature	(°C)	-20 – +60
Weight	(gm.)	1

*Value applying at rated voltage





1 Hz

5 Hz

50 Hz

500 Hz

4000 Hz

Buzz – lastname2.asm

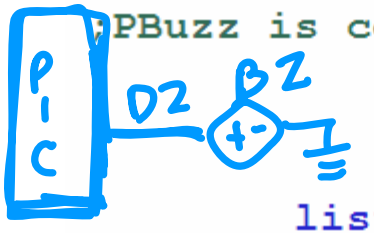
100 us Delay and 1 ms Delay

- 100 us delay
  - needs 500 instruction cycles
  - 600 =166*3 +2
    - Number of Loops=166=0xA6
  - or =165*3 +5
    - Number of Loops=165=0xA5
  - or =164*3 +8
    - Number of Loops=164=0xA4

```
;100us delay needs 500 inst
;  500 =166*3 +2 ---->Kount
;  or  =165*3 +5 ---->Kount
;  or  =164*3 +8 ---->Kount
Delay100us
    banksel Kount100us
    movlw   H'A4'
    movwf   Kount100us
R100us  decfsz  Kount100us
    goto    R100us
    return
```

```
;
Delay1ms
    banksel Kount1ms
    movlw   0x0A    ;10
    movwf   Kount1ms
R1ms    call    Delay100us
    decfsz  Kount1ms
    goto    R1ms
    return
```

Piezo buzzing Practice




```
;PBuzz is connected to RD2

        list P = 16F877

STATUS  EQU 0x03
PORTD   EQU 0x08
TRISD   EQU 0x88
PBUZZ   EQU 0x02

    CBLOCK  0x20
        TEMP
        TEMP2
        Kount120us
        Kount100us
        Kount1ms
        Kount10ms
        Kount100ms
        Kount500ms
        Kount1s
        Kount10s
        Kount1m
    ENDC
```

Dec

Variables

```
;====================
    org     0x0000
    GOTO    START

;====================

    org 0x05
START
    BANKSEL TRISD
    movlw   0x00
    movwf   TRISD

    BANKSEL PORTD
    clrf    PORTD
```

Auto Bank Selection

Example

```
    movlw   0x08        ;8 pulses of 5Hz
    banksel TEMP
    movwf   TEMP
LOOPb   bsf PORTD, PBUZZ
    call    Delay100ms
    bcf PORTD, PBUZZ
    call    Delay100ms
    decfsz  TEMP
    goto    LOOPb
```

LED-BUZZ-MOTION Practice

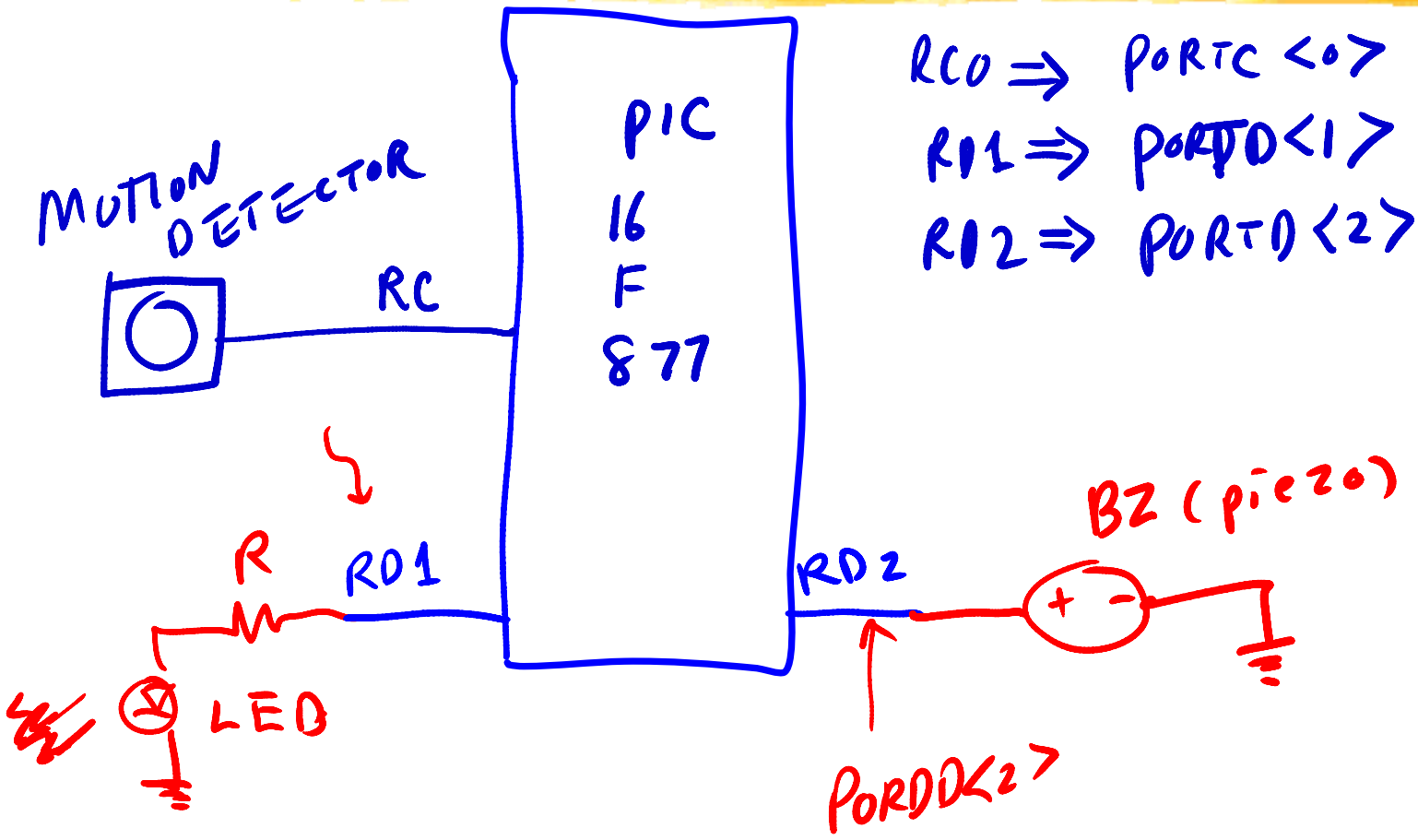
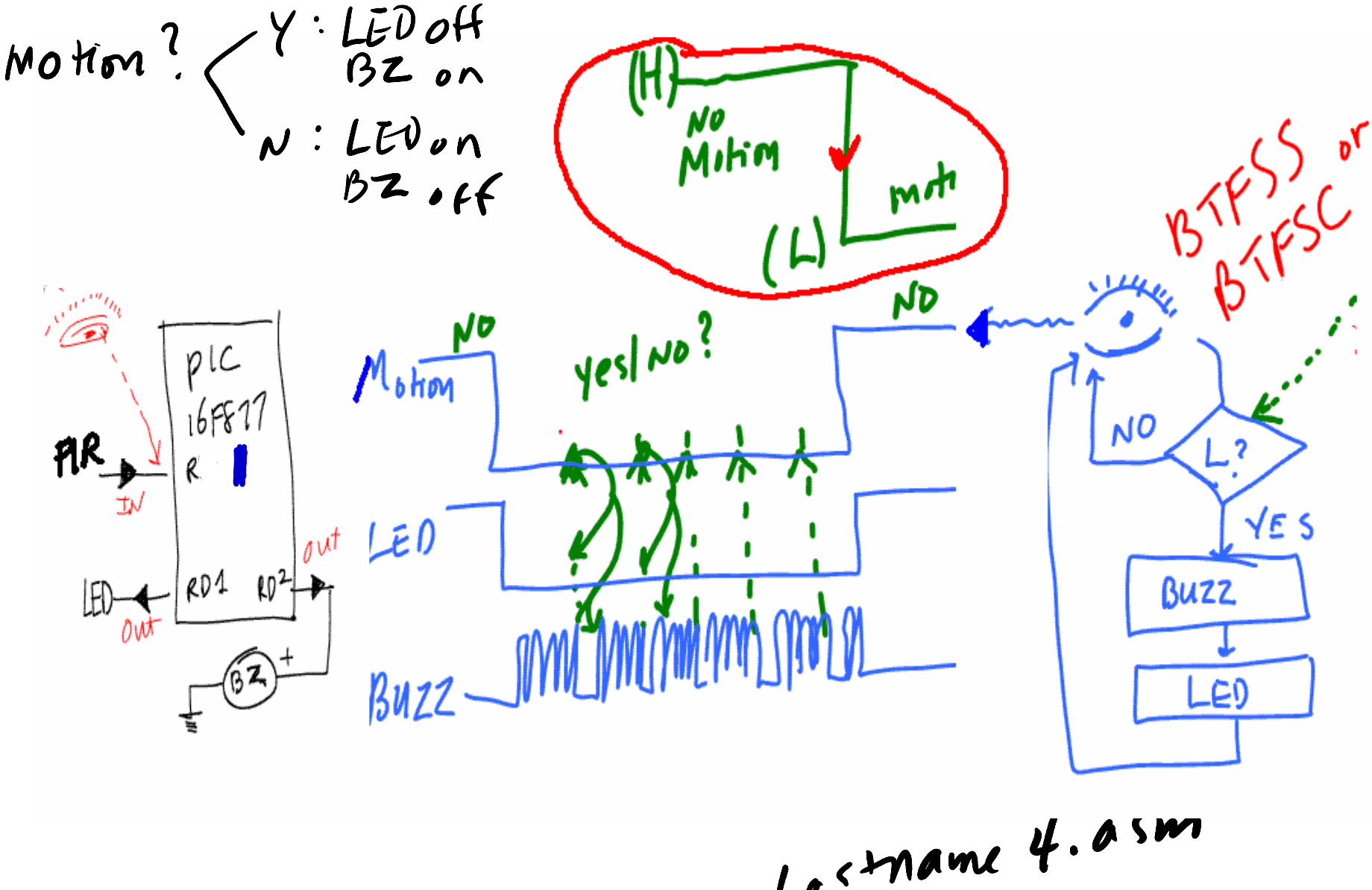
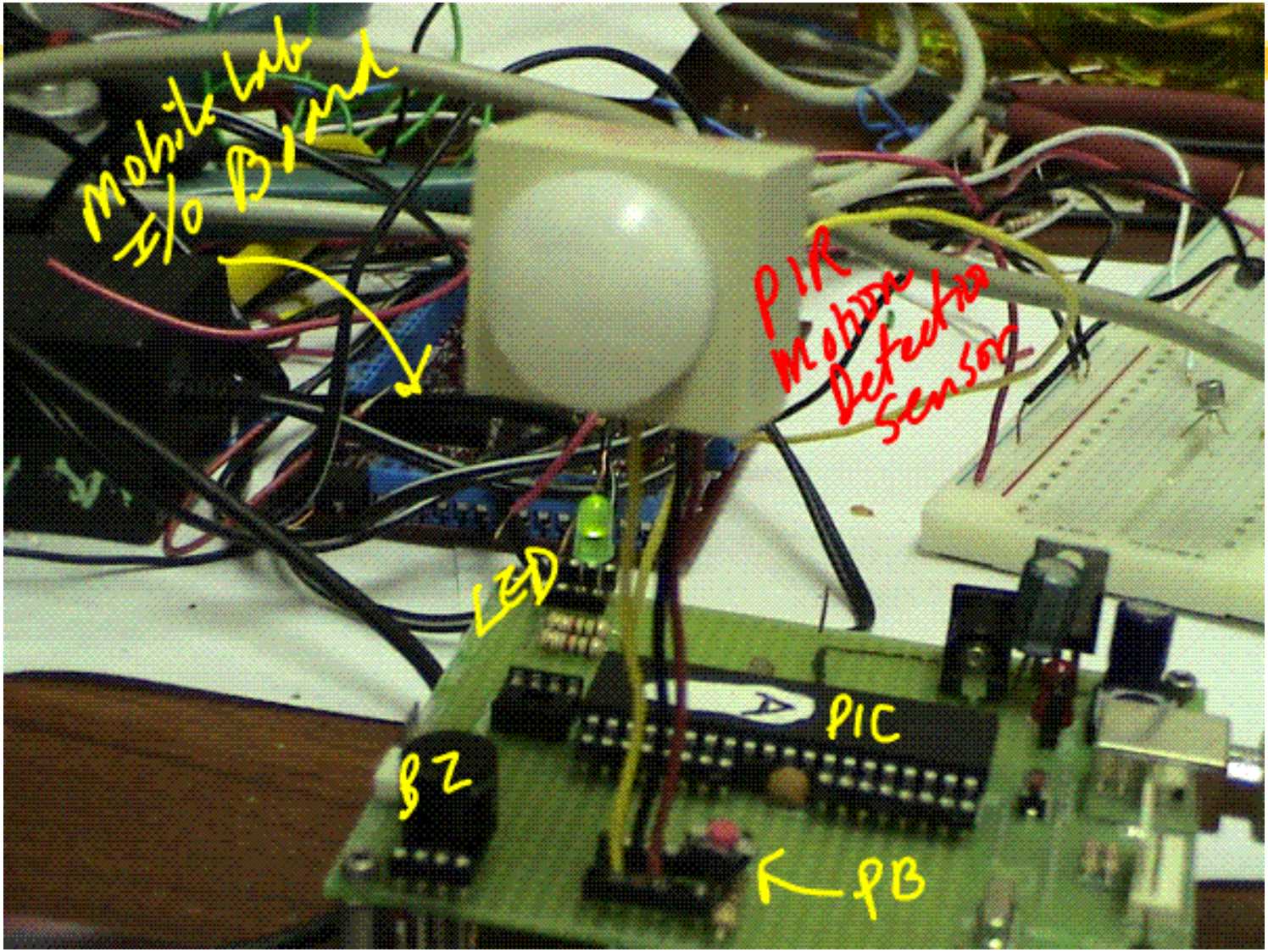


Illustration for Coding



1 < name 4.asm

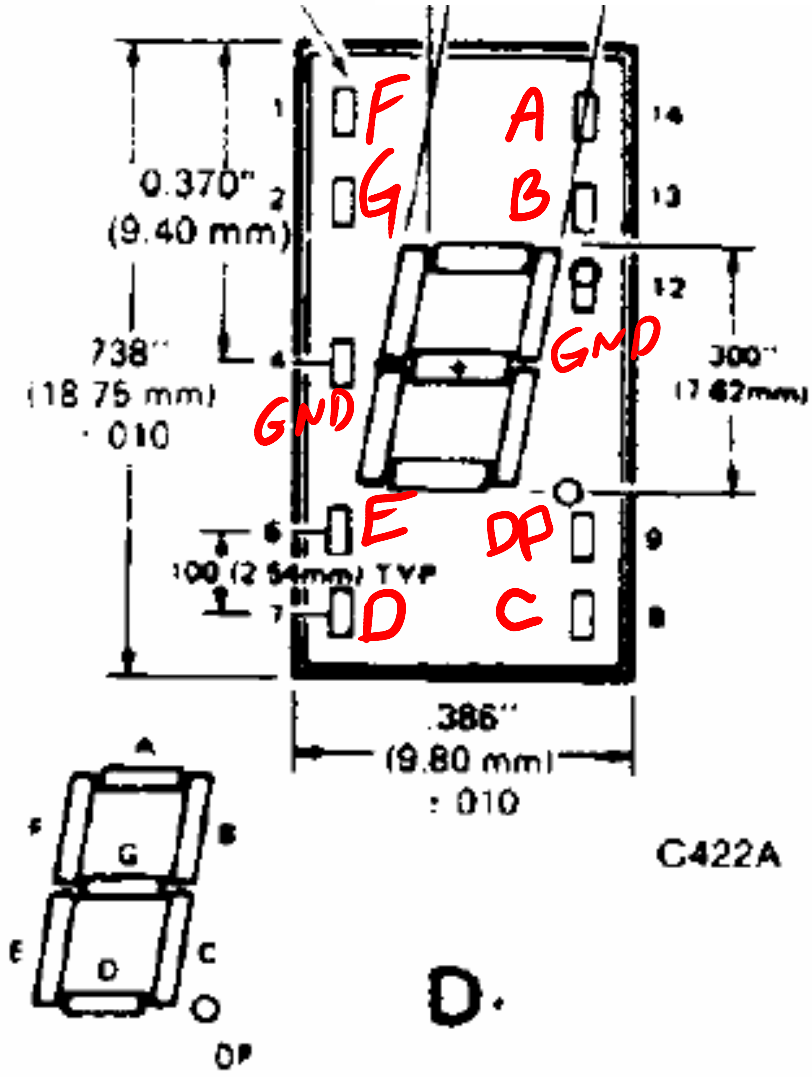
LED-BUZZ-MOTION (photo)



Coding Practice – 7 Segment LED

FAIRCHILD
SEMICONDUCTOR™

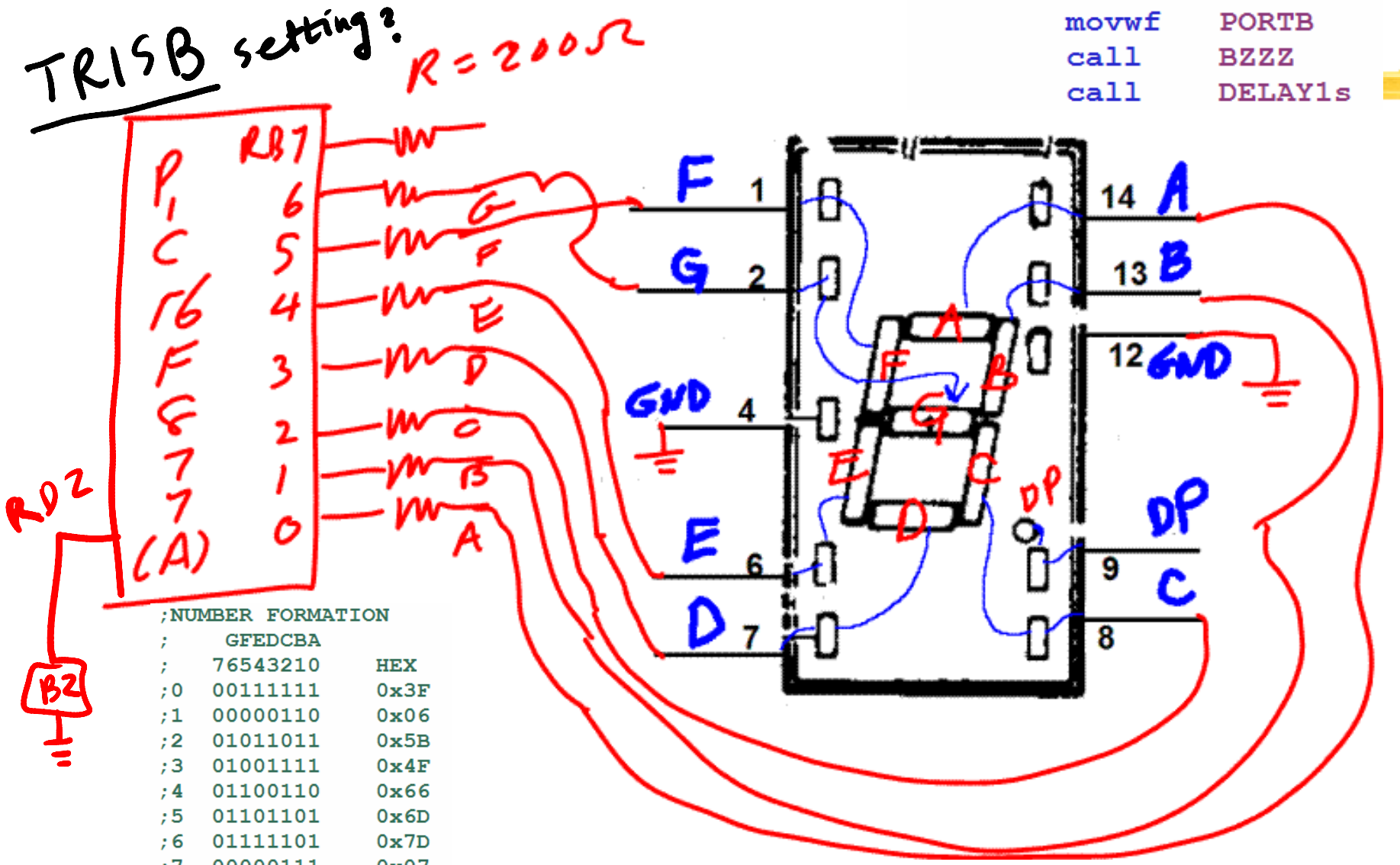
HIGH EFFICIENCY GREEN **MAN3400**
ORANGE **MAN3600**



- MAN74A
  - Common Cathode
  - +5V
  - 25mA

7 Segment Connection

```
movlw   0x3F
movwf   PORTB
call    BZZZ
call    DELAY1s
```




```
;NUMBER FORMATION
;     GFEDCBA
;    76543210     HEX
;0   00111111     0x3F
;1   00000110     0x06
;2   01011011     0x5B
;3   01001111     0x4F
;4   01100110     0x66
;5   01101101     0x6D
;6   01111101     0x7D
;7   00000111     0x07
;8   01111111     0x7F
;9   01100111     0x67
```

7 Segment Coding

7-Segment Coding Practice

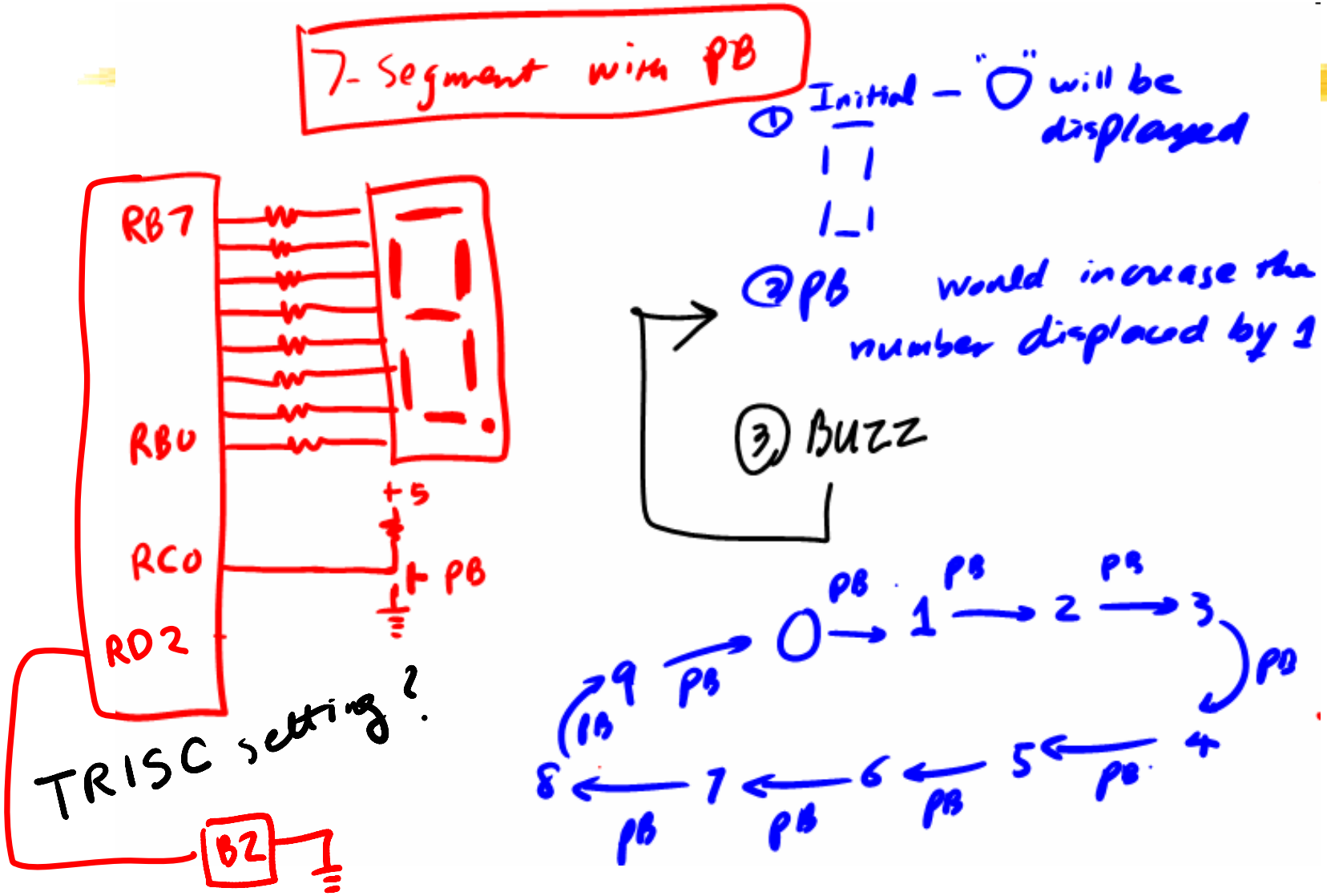
Usual Way

```
AG    movlw   0x3F   ;"0"
      movwf   PORTB
      call    Delay100ms   <Buzz
      movlw   0x__   ;"1"
      movwf   PORTB
      call    Delay100ms   <Buzz
      movlw   0x__   ;"2"
      :                    <
      :
      goto    AG
```

"Table" Way

```
AG    movlw   0x00
      call    Table
      movwf   PORTB
      call    Delay100ms
      movlw   0x01
      call    Table
      :
      goto    AG

Table         PCL
      retlw   0x3F ; "0"
      retlw   0x__ ; "1"
      retlw   0x__ ; "2"
      :
      retlw   0x__ ; "9"
```

7 Segment with PB

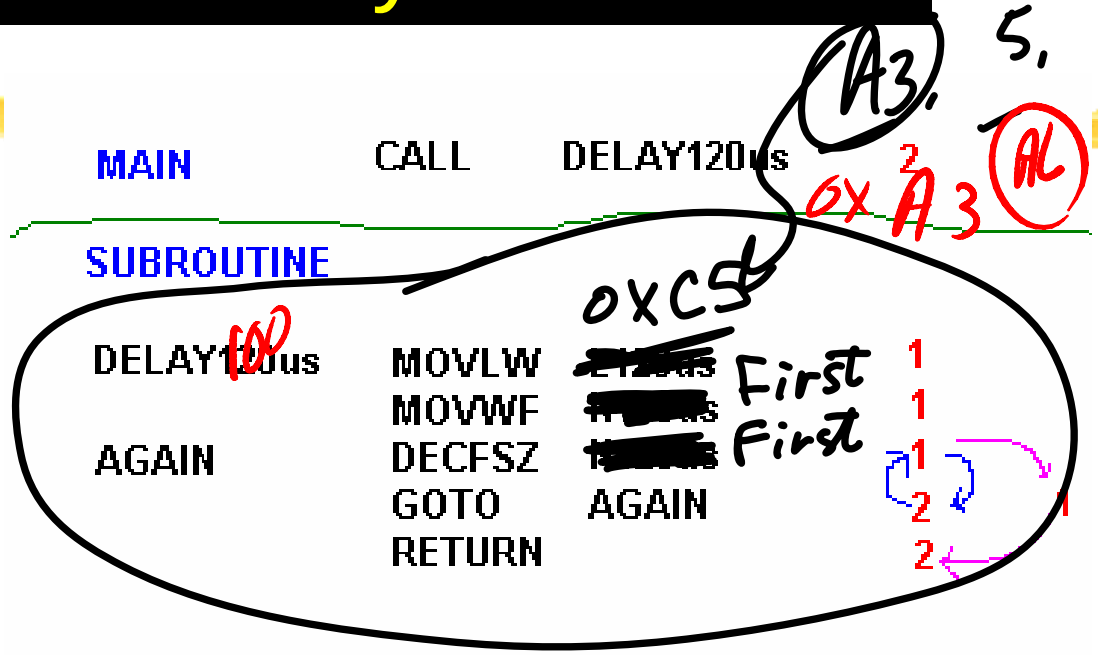


Instruction Cycles

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into standby mode	1	00 0000 0110 0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

120 us Delay

- 120 us delay
  - needs 600 instruction cycles
  - 600 =199*3 +3
    - Number of Loops=199=0xC7
  - or =198*3 +6
    - Number of Loops=198=0xC6
  - or =197*3 +9
    - Number of Loops=197=0xC5



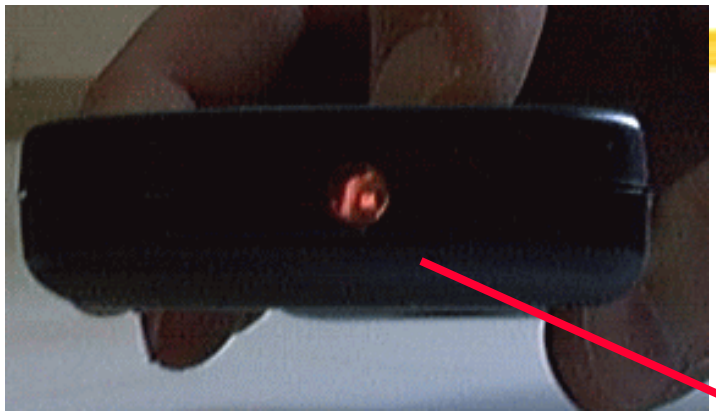
CALL (2) + MOV (2) + n120us*(3)+1(extra)+RETURN(2)
= (7 + n120us*3) instrcution cycles

Delay 10ms

Again 2

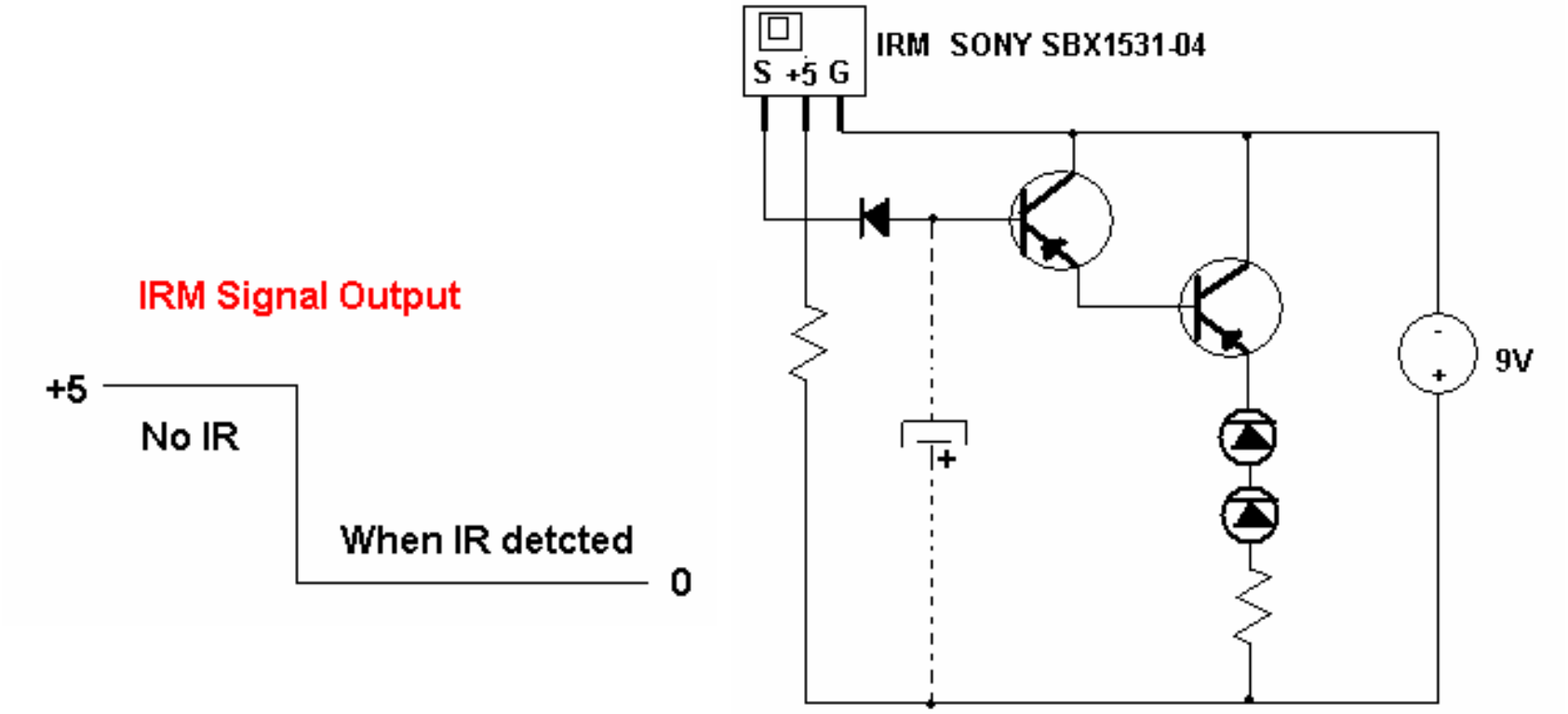
MovLW #6A 64
movwf second
decfsz second
Call Delay 100us
goto Again 2
return

5: Infra Red Control – Sony Remote



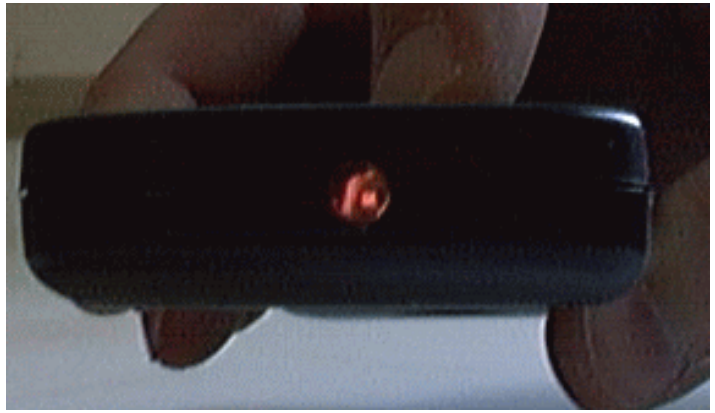
Simple IR Application

- IR Remote Control Night Light



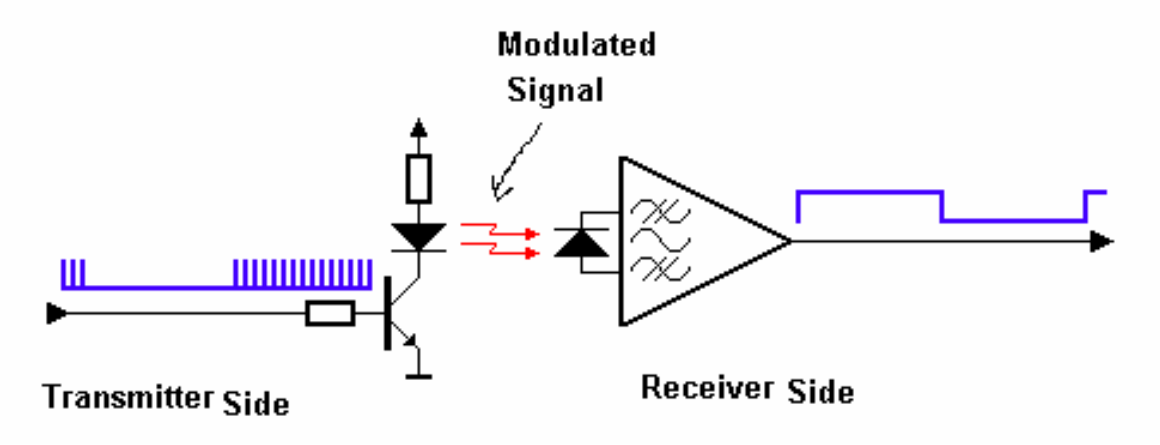
IR Control

- Infra-Red light: cheapest way to remotely control a device within a visible range
- Almost all audio and video equipment are now controlled by IR
- IR Protocols
  - Sony
  - Sharp
  - Philips



IR Modulation

- Modulation: To make signal stand out above the noise.
- With modulation we make the IR light source blink in a particular frequency. (30 – 60 KHz)
- The IR receiver will be tuned to that frequency, so it can ignore everything else.

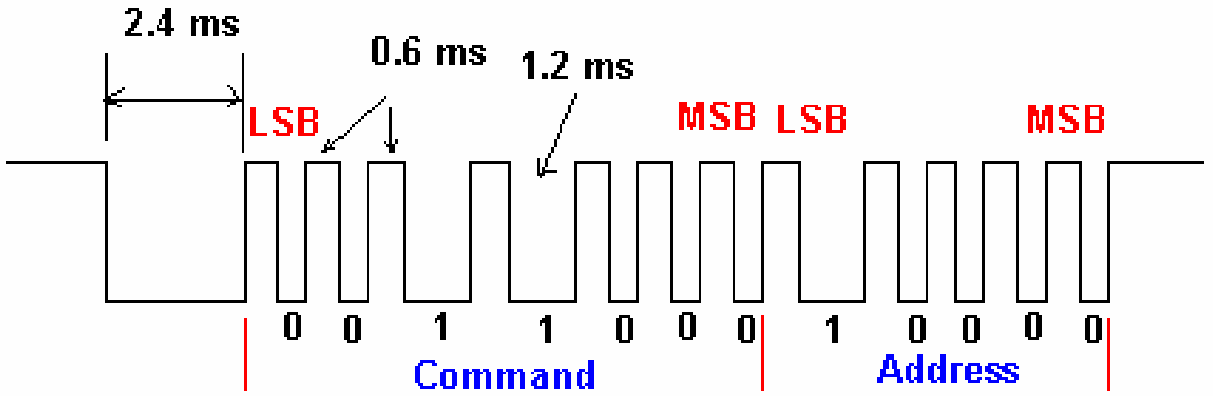


Sony Protocol –Addr/Com

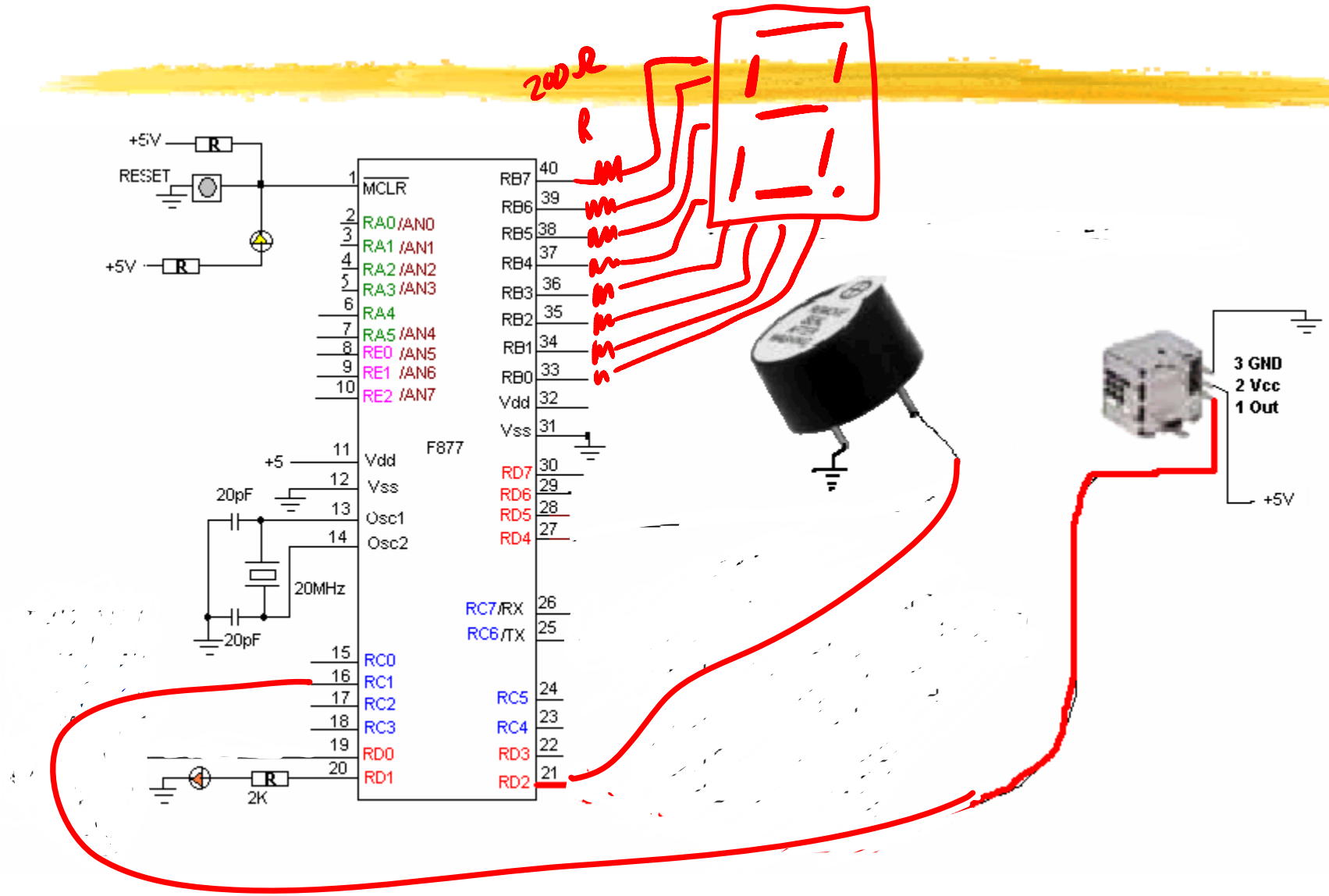
- **Address**
  - **1: TV**
  - **2: VCR1**
  - **3: VCR2**
  - **6: Laser Disk Unit**
  - **12: Surround Sound**
  - **16: Cassette Deck/Tuner**
  - **17: CD Player**
  - **18: Equalizer**
- **Command:**
  - **0 – 9: Keys 1 – 0**
  - **16: Channel +**
  - **17: Channel –**

SONY Protocol

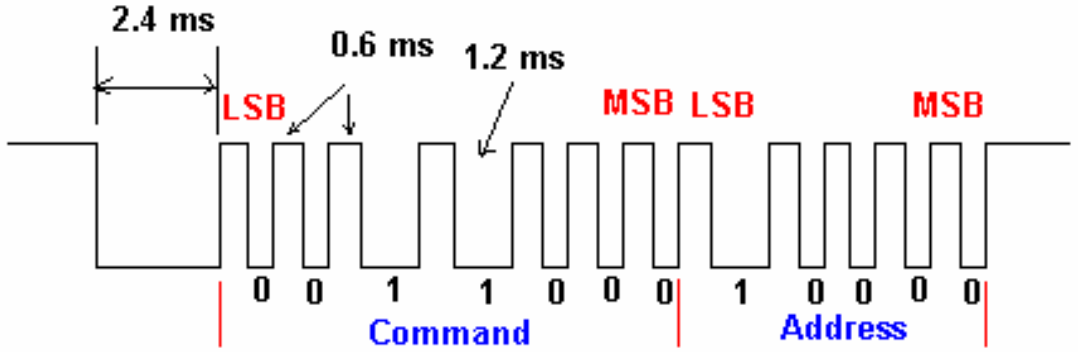
- 12-Bit of Information
- 5-Bit for Address and 7-Bit for Command
- Pulse Width Modulation
- Carrier Frequency 40 KHz
- Bit Time: 0.6 ms (0) or 1.2 ms(1)
- Commands are repeated every 45 ms as long as a key is held down.



IR connection Diagram

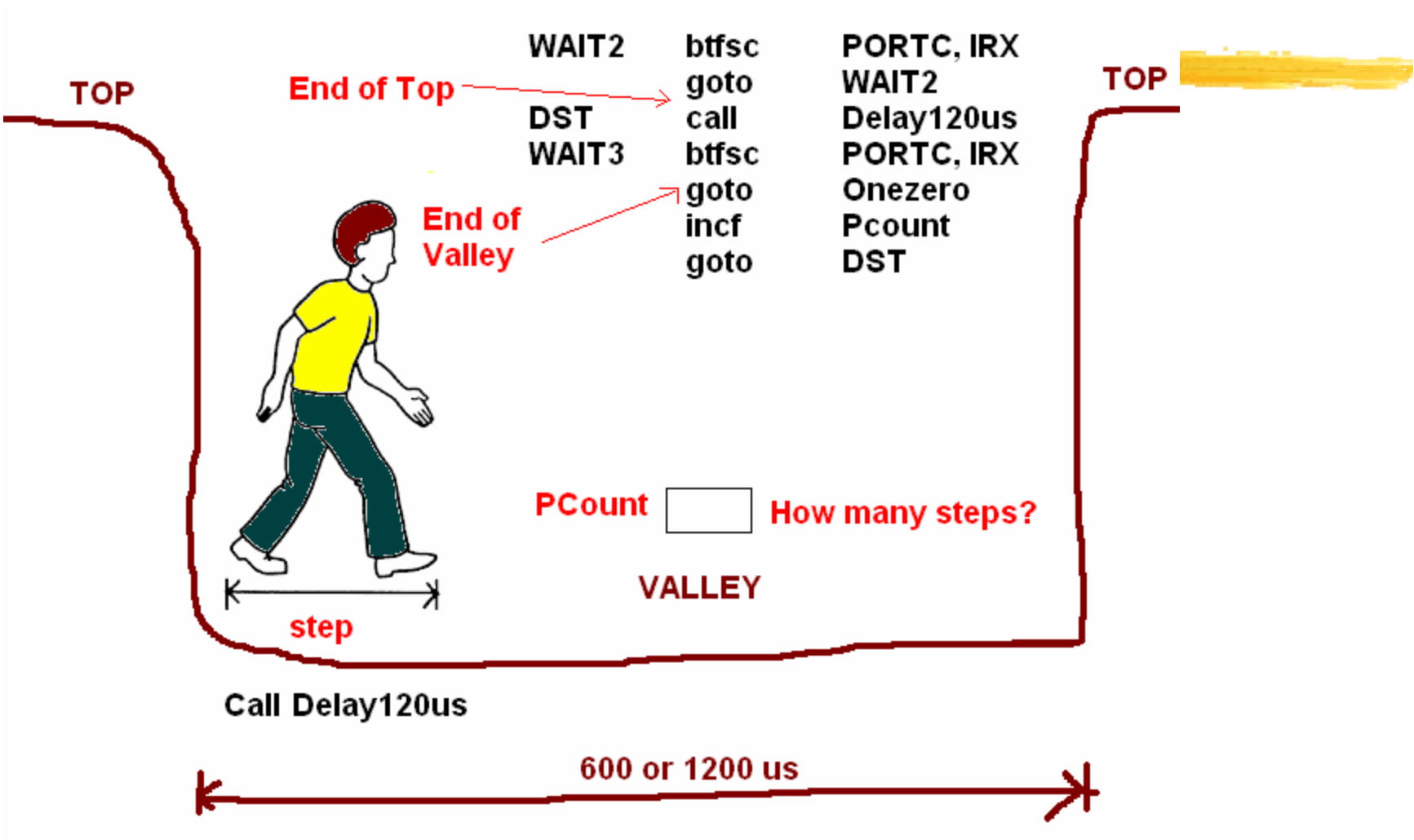


Sony Protocol –Bit Reading Scheme

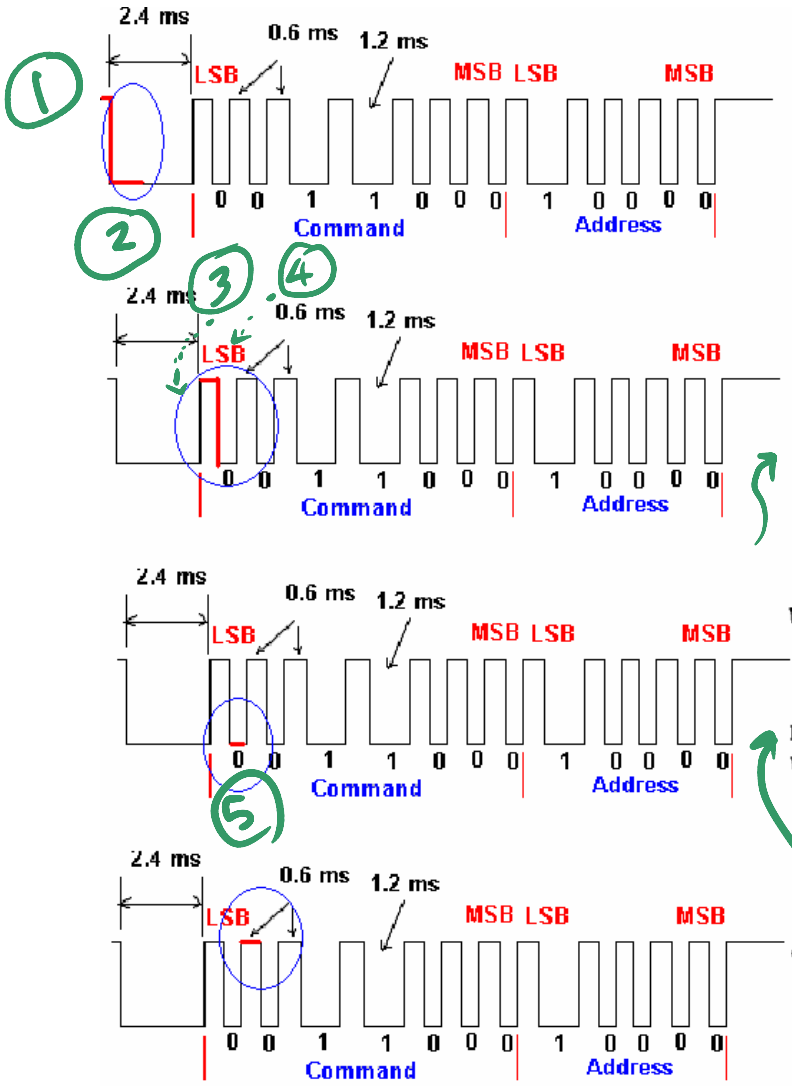


- "1" : 1200us
- "0": 600 us
- Sequence
  - 1. Detect IR for LOW (START)
  - 2. Wait until IR goes to HIGH (Separator)
  - 3. Wait until IT goes to LOW
  - 4. Wait for 120us
  - 5. Check IR if it goes to HIGH
    - If Not, Increase a counter by 1 and go to 4
    - If High
      - Count<8: "0"
      - Count>8: "1"
      - Go to 3 (to read next bit information)

Measuring the pulse width



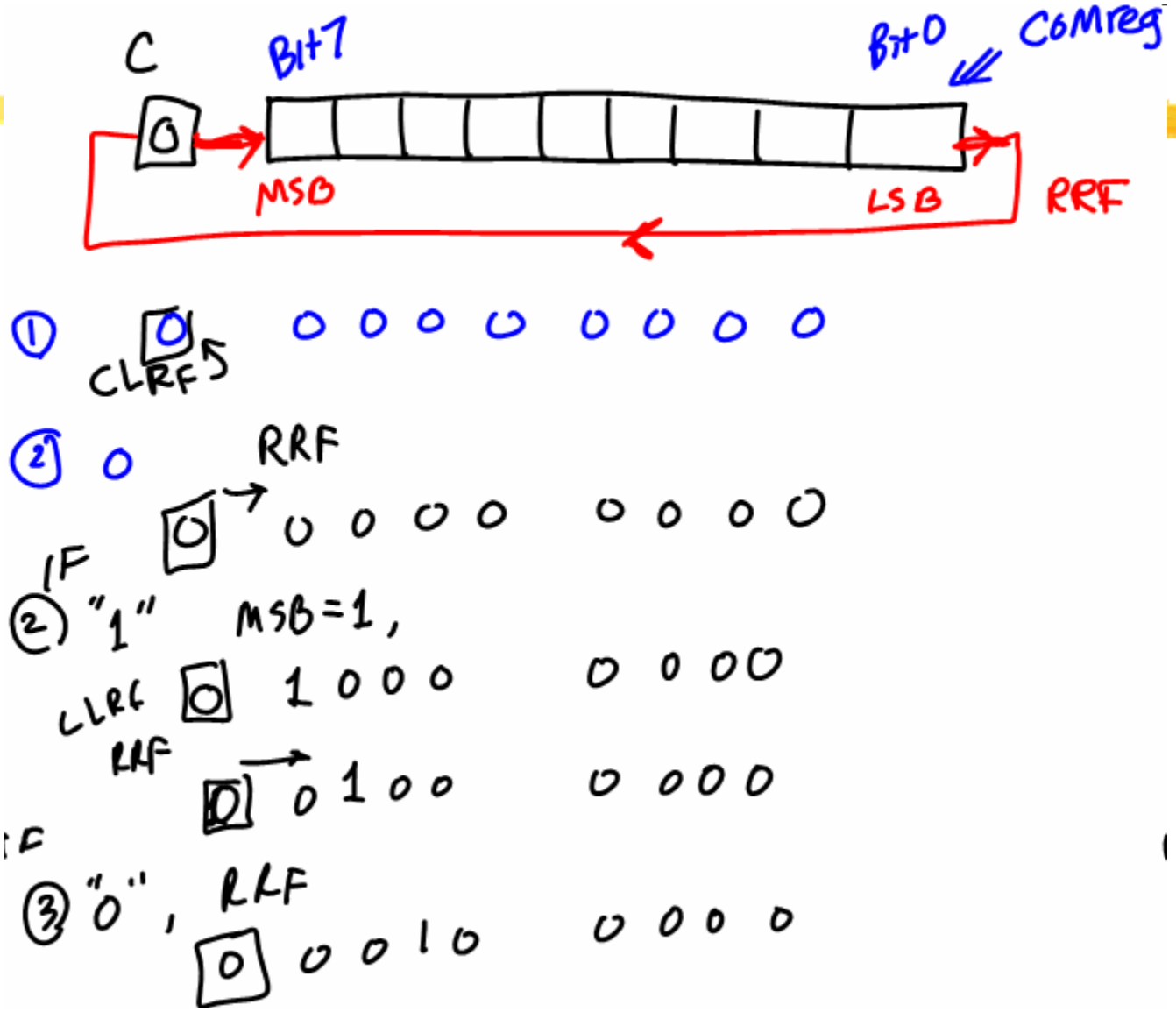
Sony Protocol – Coding example for COMMAND reading



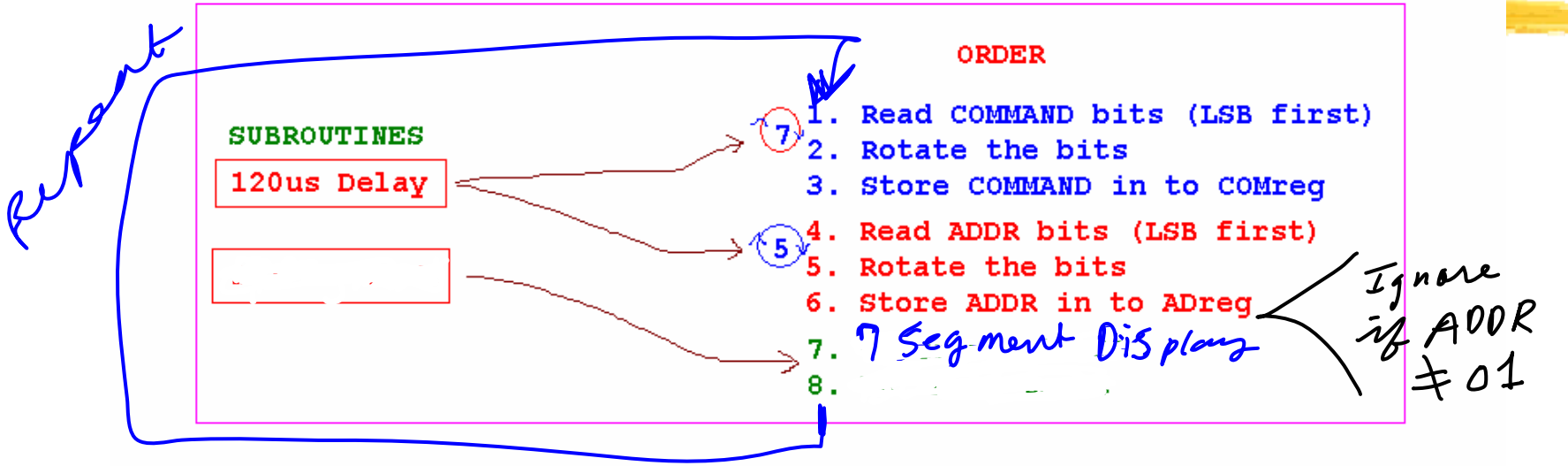

```
; START OF COMMAND READ
;1. Wait for START bit
JAM
        banksel PORTD
        btfsc   PORTD, IRX      ;IRX=2
        goto    JAM
;2. Once START is entered
        banksel CMcount
        movlw   0x07    ;Command has 7 bits
        movwf   CMcount
;3. Wait for separator (600us length)
WAIT    btfss   PORTD, IRX
        goto    WAIT
CMNEXT  clrf    Pcount  ;Number of 120us duration
        bcf     STATUS, CARRY
        rrf     COMreg          ;storage for COMMAND
                                ;MSB is 0 NOW
;4. WAIT for the end of separator
WAIT2   btfsc   PORTD, IRX
        goto    WAIT2

;5. Pcount update (count how many 120us Low duration)
DST     call    delay120us
WAIT3   btfsc   PORTD, IRX
        goto    Onezero         ;End of LOW duration
                                ;1 or 0 ?
        incf    Pcount
        goto    DST
;6. At the end of LOW duration
Onezero btfsc   Pcount, 0x03    ;What is this for?
        bsf     COMreg, MSB     ;the MSB is now 1
        decfsz  CMcount
        goto    CMNEXT
;7. Once all 7 bit inforation read
        bcf     STATUS, CARRY
        rrf     COMreg    ;rotate one more for 8-bit re

;END OF COMMAND READ
```

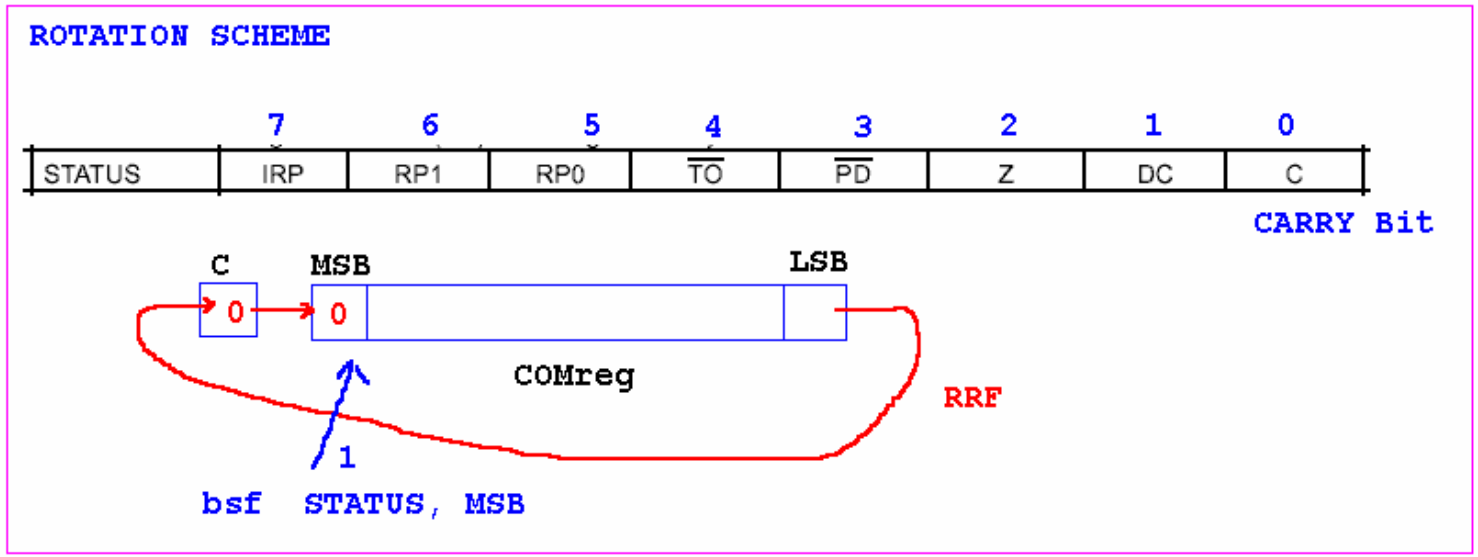
PORTD → PORTC

Correcting the Order (to 8-bit information)

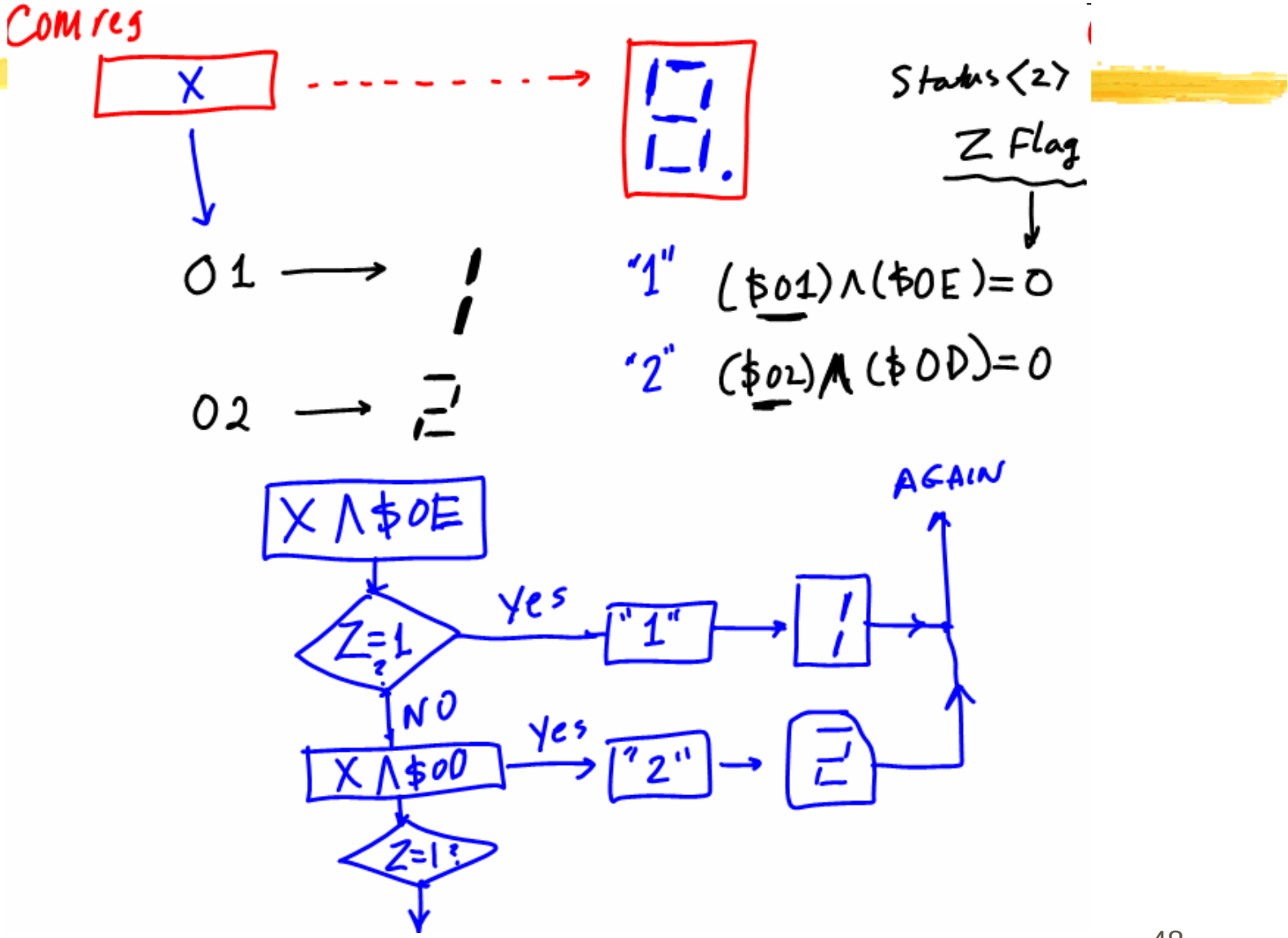


IR Coding Structure





COMreg → 7 Segment



Testing