

Ass 1 final

by Morgan Jones

FILE	3676_MORGAN_JONES_ASS_1_FINAL_153432_1236147062.DOCX (2.95M)		
TIME SUBMITTED	31-JAN-2019 03:17PM (UTC+0000)	WORD COUNT	5530
SUBMISSION ID	100071038	CHARACTER COUNT	31289

Name: Morgan jones

Student number:

1706863

Module :

(ACDA4004)

Course: Electric and

Electronic

Engineering

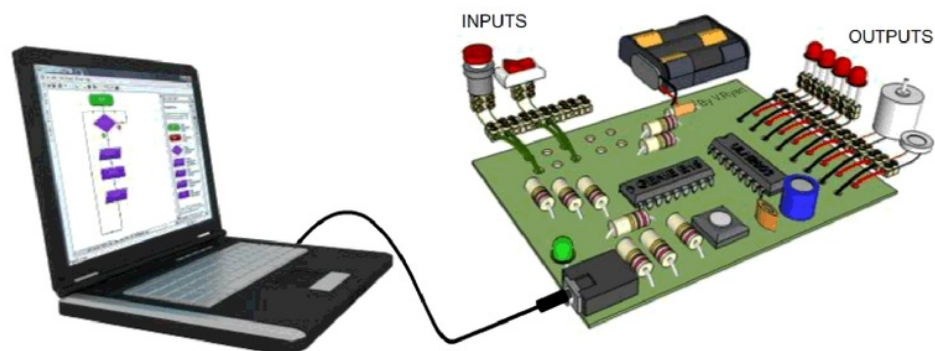
Contents

Name: Morgan jones	1
Student number: 1706863.....	1
Module : (ACDA4004).....	1
Course: Electric and Electronic Engineering.....	1
Introduction	2
Section 1:	14
Section 2:	17
Bibliography.....	21
Appendices	22

Introduction

What is a PIC ?

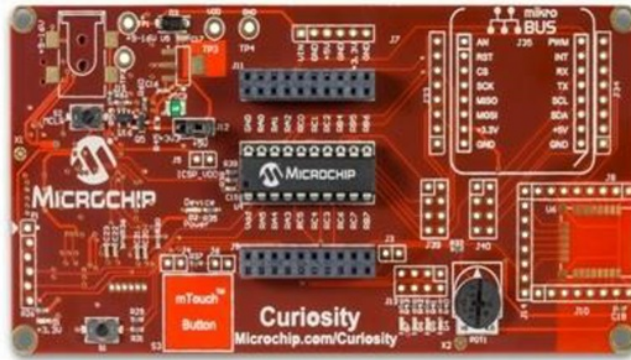
PIC microcontrollers is an electrical circuit that is set to bring out a variety of tasks. They can be used in in things from clocks to manage a construction line. They can be found in many electronic devices such as alarms, computer systems and phones in essence they can be in any electronic device. [1]



[1]

There are many types of pic board, here are some:
Curiosity pic microcontroller Development board

The curiosity program development board was created by microchip for 8-bit pic microcontrollers. Its figure is shown below:



Features of Curiosity pic Development board:

1. It supports the 40 pin 8-bit microcontroller with low voltage.
2. It has USB interface with integrated programmer or debugger.
3. It has RN 4020 Bluetooth foot print.
4. It has physical switches and analogue potentiometer.
5. It has wide interface options such as m touch button.

Ready for PIC microcontroller development Board

Reay of pic is the development board which is used for development of 28 and 40 pins pic microcontrollers. It is shown in figure below:



Features of Reay for PIC development Board:

1. It is fully supported by mikro c mikro basic and mikro pascal compiler for pic compiler.
2. It has 10 IDC headers for available microcontroller ports.
3. It has USB-UART module, prototyping area and power supply circuit.
4. It has fast UART bootloader for on board pic programming.

XLP 8-Bit pic microcontroller Development Board

XLP 8-Bit Development Board is designed by microchip for a very low power and have a very low sleep current. microcontrollers. This XLP 8-Bit Development Board is shown below.



Microchip XLP 8-Bit Development Board

Features of XLP 8-Bit pic microcontroller Development Board:

1. It supports the Pic 18F87k22, which have 80 pins and 128KB flash memory.
2. It also supported the microcontroller such as 6LF194PIC17 which 64 pins and 28KB flash memory.
3. It also has current and voltage measurements facility.
4. It has USB interface for power and PC communication.
5. It has LEDs for identifications.
6. It has LCD interface facility.
7. It has analogue temperature sensor and CTMU based diode temperature sensor.
8. It has power options such as energy harvesting, AAA, USB, CR2032, external or 9 V power supply.
9. It has very easy connection with MPLAB PICKIT3, ICD3 or RESL ICE for input debugging and programming

XLP 16-Bit pic microcontroller Development Board

XLP 16-Bit Development Board is also designed by microchip and is used for very low power applications such as low power security systems and low power motor control. This XLP 16-Bit Development Boards shown below.



Features of XLP 16-Bit pic Development Board

1. It has 20 and 28 pins shared footprints sockets with associated headers for PIC 24F microcontroller.
2. It has 8 MHz and 32.768KHz oscillator circuit for PIC 24F microcontroller.
3. It has power LED.
4. It has power control switch IC (IC PWR).
5. It has adjustable regulator (LDO).
6. It has temperature sensor, potentiometer and serial EEPROM.
7. It also has capacitive touch pads and capacitive measurements points

XLP 32-Bit pic microcontroller Development Board

XLP 32-Bit Development Board is also created by microchip, can provide the lowest and easiest method to experience the 32 bit pic microcontrollers.



Features of XLP 32-Bit pic microcontroller Development Board

1. It has free MPLAB IDEX and MPLABXC32 compiler.
2. It has 512K flash, 32K ram and 4 channel DMA. USB powered board.
3. It has 3 users switched and 3 LEDs.
4. It has full source based debugging in MPLAB XIDE.
5. It run and verify the program in 9 seconds.

Easy PIC V7 8-Bit pic microcontroller Development Board

This board is created by mikroelektronika development tools and is designed for experienced the 8-bit microcontroller such as PIC 18F78K22. It is shown in figure below:



Features of Easy PIC V7 8-Bit pic Development Board

1. It could be used for variety of applications such as Ethernet, USB, temperature measurements and including graphics.
2. It also supports the PIC18F, PIC18FJ, and PIC18FLK.
3. It has the special function registers (SFR).

4. It has CODE, RAM, and EEPROM along with code execution hardware.
5. It also 80 Pins MCU card that could be sold separately.

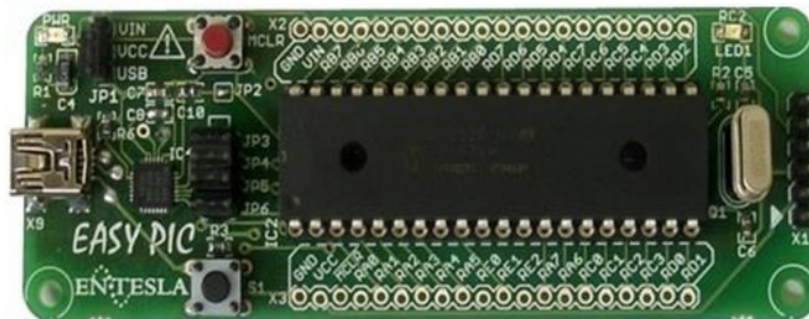
PIC KICK Development Board with PIC16F877A

This is created for pic 16F and 18F microcontrollers. It is shown in figure below



EASY PIC Development Board for PIC18F4520

This is for pic 18F4520 i]] board for 40 and 28 pin microcontrollers. it is shown below:



Features of EASY PIC Development Board for PIC18F4520:

1. It has serial a bootloader with on board microchip 18F4520 microcontroller.
2. It is powered up externally or USB port.
3. It has on board USB to serial communication converter
4. It 12 MHZ replaceable crystal to clock the microcontroller.
5. It also has reset push button.

PIC-P26J50 Porto pic Development Board

This is created for the pic 18F26J56 microcontroller and built with high speed low power CMOS flash memory technology. It is shown below:



Features of PIC-P26J50 Porto Development Board:

1. It has 64 K flash with pic microcontroller.
2. It has UBS connector with 8 LEDs.
3. It has user button and reset button.
4. It has UEXT connector.
5. It has dimensions 61*45mm.

[2]

what the purpose of MPLAB is, and what are its general features.

MPLAB is produced by Arizona Microchip. MPLAB IDE is an incorporated toolset for the enhancement of fixed applications on PIC Microchips and microcontrollers. It is called IDE because it provides a single integrated environment to create code for embedded microcontrollers.

Features

MPLAB IDE is a Windows OS based IDE for the PIC families and Signal Controllers. MPLAB provides the capability to build and rewrite source code via the incorporated editor.

Build, assemble and connect source code.

Debug the executable logic by watching programs flow with the incorporated simulator or in real time with in-circuit emulators or in-circuit debuggers.

Make scheduling measurements through the simulator or emulator.

[3]

Circuit Designing

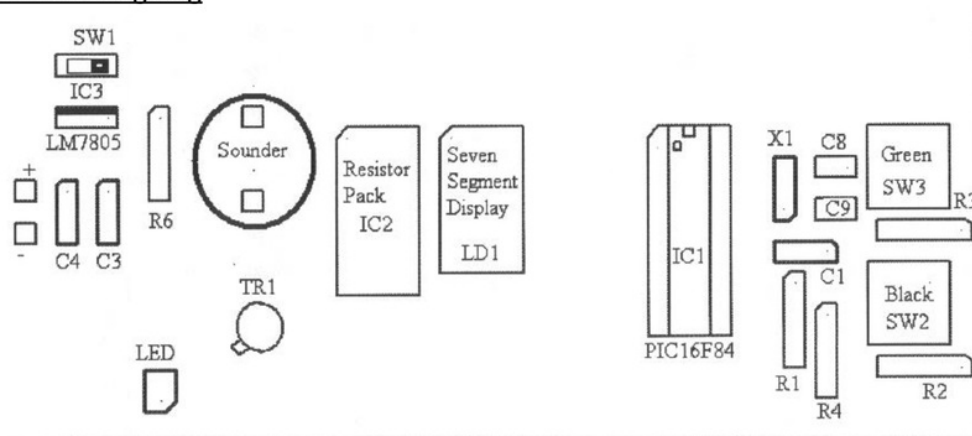


Figure 1 diagram of the physical device

The PIC microcontroller transmits as well as receives data with electrical pulses, the PIC can operate with a crystal that has a set **MHz**. The capacitors are attached to a crystal oscillator which is utilized to steady the electrical signals.

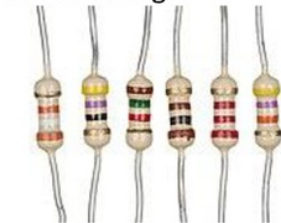
Circuit Components

Reset device, A power-on reset generator is a microcontroller that generates a reset signal when power is applied to the device. [4]

Capacitors are utilized in energy supplies where they soften the output.



Resistors are employed to lessen current flow, modify signal levels and divide voltages.



As you can see the diagram shows the physical components on the PIC board there are two buttons one black and one green these can be used to control the actions of the sounder or seven segment display.

EC Buying



Buttons [7]

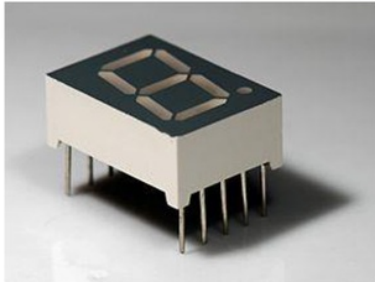
There is a switch which helps control whether the board receives power in essence turn on or off there is also a led connected to the switch which helps verify that there is power going to the components.

There is a sounder this is used to create a sound for example if the pic is programmed so that the sounder creates a tune or plays a piece of music



Piezo Buzzer [8]

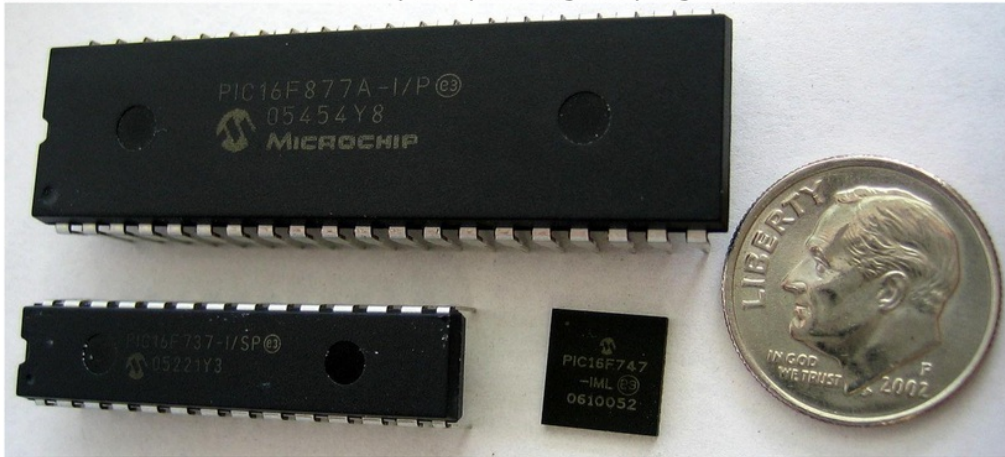
There is a seven segment led display this is used to show any particular letters or numbers that the pic has been program to show for example "H" "E" "L" "P" or "S" "T" "O" "P"



Seven segment display [9]

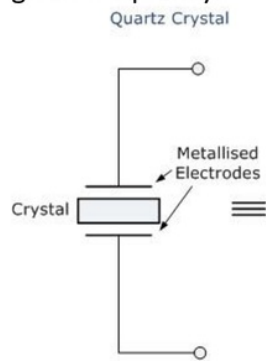
There is also the PIC microcontroller which can be connected and disconnected from the board mean it is easier program with a set of instructions as if it was stuck on you would not be able to program the

PIC as there will be feasible way or uploading the program code.



[10]

There is crystal oscillator which is used for the resonance of a vibrating crystal of piezoelectric material. It will create an electrical signal with a given frequency.



crystal oscillator

[11] [12]

Section 1:

Error 1

Error[113] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 30 : Symbol not previously defined (status)

Error[113] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 31 : Symbol not previously defined (status)

Error[113] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 33 : Symbol not previously defined (status)

Error[113] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 36 : Symbol not previously defined (status)

As you can see from the above lines there are multiple line with the same error these lines however are not the issue line 17 is as there is a mistype as the word Status should be lowercase not uppercase as it is in the count file an easy way to fix this is go to line 17 and change the word to lowercase letters.

Error 2

Error[113] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 57 : Symbol not previously defined (get_count)

The above error is not with line 57 specifically but line 41 as get_counnt should match line 57 however due to a mistype there is a discrepancy of get_count having an extra n in the word count to rectify this just remove the extra n.

Error 3

Warning[202] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 74 : Argument out of range. Least significant bits used.

The error for this line is another mistype as the line shows movlw 0xffff When it should show movlw 0xff as this indicates what is show on the segment board as 0xffff will either make the program not work or cause it to act different to the programmer's intentions so to fix this issue remove an f making it 0xff

Error 4

Error[113] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 75 : Symbol not previously defined (mCont)

The error for this line is that there is a mistype as the word mcont this is not the correct instruction and should really be mcount as it works as the outer loop counter and to correct it you should add a u in between the o and n making the correct instruction

Error 5

Warning[207] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 89 : Found label after column 1. (en)

The error that the full word should be end instead of en but there was a mistype as there was a gap between the en and d so this can be corrected by removing the gap making the word end instead of en d

Error 6

Error[122] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 89 : Illegal opcode (d)

The error that the full word should be end instead of d but there was a mistype as there was a gap between the en and d so this can be corrected by removing the gap making the word end instead of en d

Error 7

Error[129] C:\USERS\ERGO\DOWNLOADS\COUNT.ASM 98 : Expected (END)

This error is a mistyped word as there should be the word end instead of en d but there was a mistype as there was a gap between the en and d so this can be corrected by removing the gap making the word end instead of en d

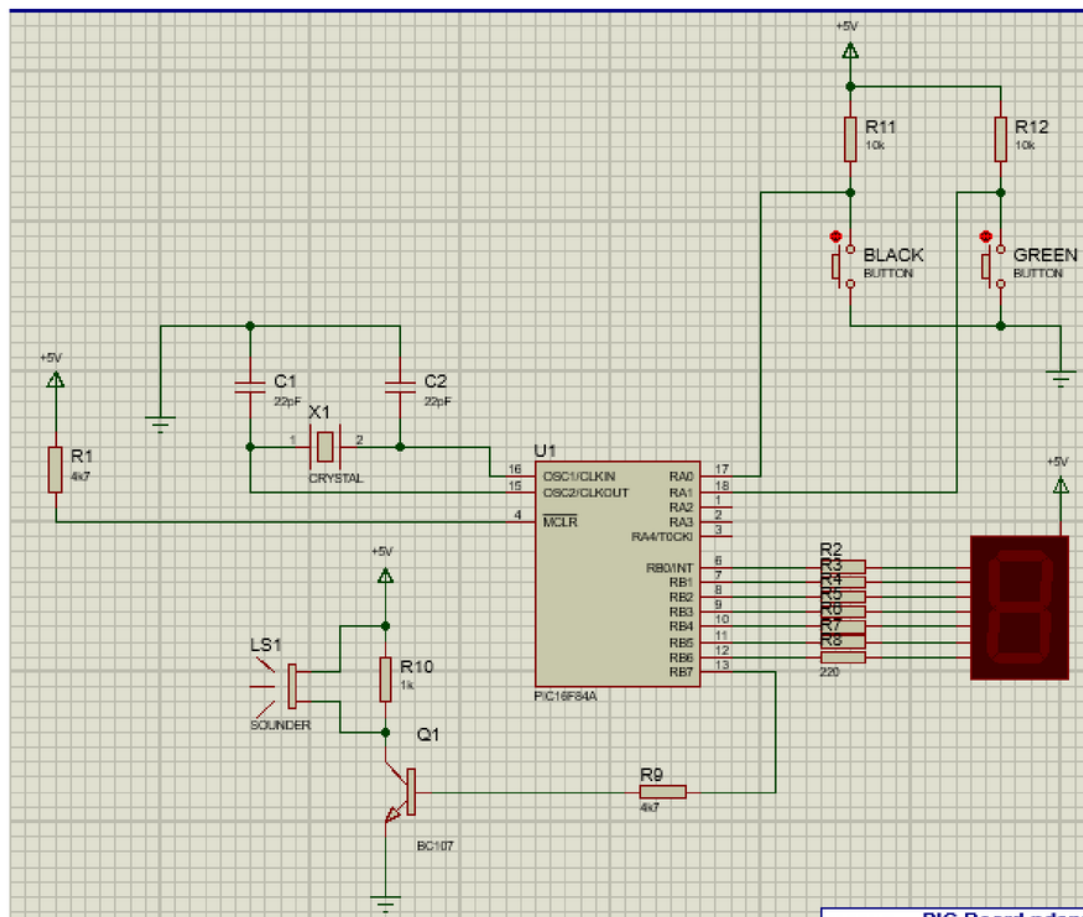


Figure 2 proteus schematic of the PIC board

Description of the purpose and general operation of each component in MPLAB.

The MPLAB possesses both fixed and plug-in components to set up the system for an assortment of software and hardware tools.

Project Manager, this offers incorporation and interaction between the IDE and the tools.

Editor is a full-featured programmer's script editor that also serves as a way to the debugger.

Assembler and Language Tools can be utilized as stand-alone to construct a single file or can be utilized with the linker to build a project from multiple source files, libraries and recompiled objects.

The linker is responsible for positioning the compiled code into memory areas of the target PIC.

Debugger: The debugger allows breakpoints, single stepping, watch windows and all the features of a modern debugger for the MPLAB. It can work in combination with the editor to reference information from the target being debugged back to the source code.

Execution Engines: There are software simulators in MPLAB for all PIC devices. These simulators use the PC to simulate the instructions and some peripheral functions of the PIC devices.

[3]

Section 2:

1. Modify "Count.asm" so the seven segment display shows "H" "E" "L" "P" at a rate that is easy to read.

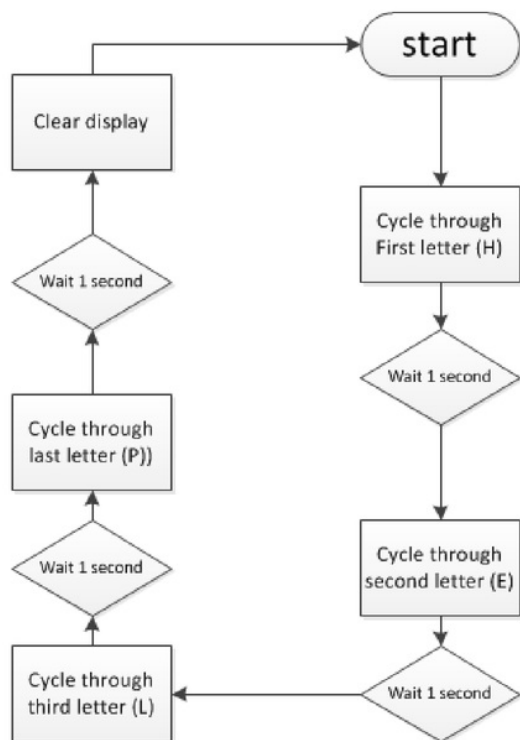


Figure 3 flowchart design for count.asm

The flowchart shows that when the power switch is turned on the PIC will light the led with the first letter H then it will wait 1 second then proceed to the next letter E then it will wait 1 second then proceed to the next letter P then it will wait 1 second then proceed to the next letter P then it will wait 1 second then proceed to clear the led display and then restart the program

Modify "Count.asm" so that the piezo sounder produces a 440Hz tone while the Green button is pressed.

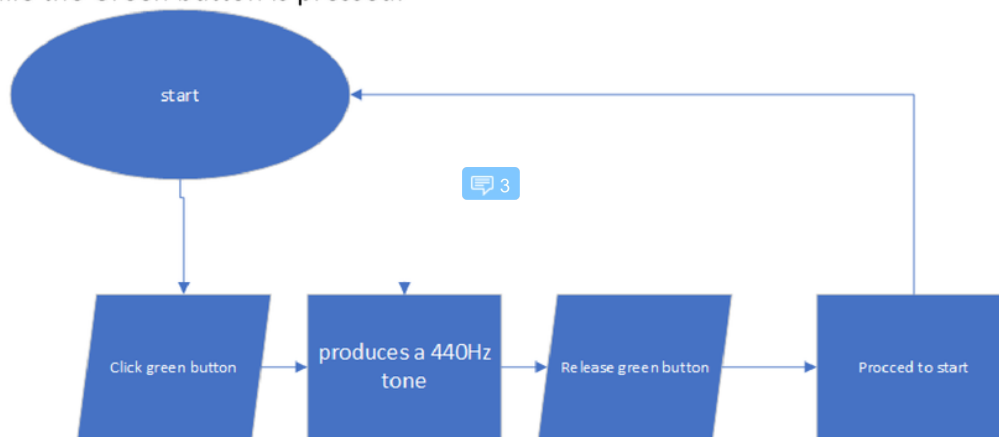


Figure 4

The flowchart shows that the when the power switch is turned on you can press the green button as the sounder produces a 440 hz tone then you release the green button then the program will reset so that you can repress the green button and the activate the sounder .

Modify "Count.asm" so that the piezo sounder produces a low frequency tone while the Green button is pressed and a high frequency tone while the Black button is pressed.

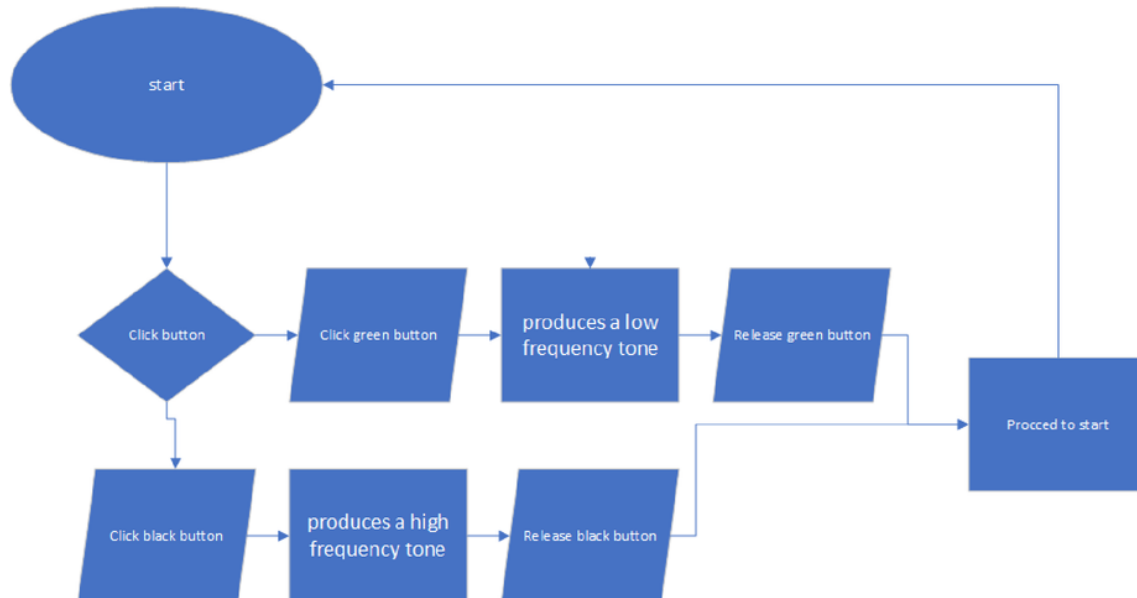


figure 5

The flowchart shows that the when the power switch is turned you will have to decide which button to press if you press the green button the sounder will produce a low frequency tone once the button is released the program will restart and if you press the black button the sounder will produce a high frequency tone once the button is released the program will restart.

Bibliography

- [1] V. Ryan, "WHAT IS A PIC MICROCONTROLLER? WHAT CAN IT DO?," [Online]. Available: <http://www.technologystudent.com/pics/picgen1.html>. [Accessed 1 january 2019].
- [2] "Pic Microcontroller Development Boards with Their Features," [Online]. Available: <http://microcontrollerslab.com/pic-microcontroller-development-boards-features/>. [Accessed 21 january 2019].
- [3] A. Tomar, "Microchip: MPLAB Integrated Development Environment (IDE) Overview," [Online]. Available: <https://www.element14.com/community/docs/DOC-39318/1/microchip-mplab-integrated-development-environment-ide-overview#anchor1>. [Accessed 1 January 2019].
- [4] "Power-on reset," [Online]. Available: https://en.wikipedia.org/wiki/Power-on_reset. [Accessed 11 January 2019].
- [5] "Applications of capacitors," [Online]. Available: https://en.m.wikipedia.org/wiki/Applications_of_capacitors. [Accessed 12 january 2019].
- [6] "Resistors," [Online]. Available: <https://en.m.wikipedia.org/wiki/Resistor>. [Accessed 12 january 2019].
- [7] "Aliexpress buttons," [Online]. Available: <http://www.aliexpress.com/i/32851421040.html>. [Accessed 11 january 2019].
- [8] "robomart.com piezo-buzzer-b-10n," [Online]. Available: <https://www.robomart.com/piezo-buzzer-b-10n>. [Accessed 11 january 2019].
- [9] "Seven-segment display," [Online]. Available: https://en.wikipedia.org/wiki/Seven-segment_display. [Accessed 1 January 2019].
- [10] "PIC microcontrollers," [Online]. Available: https://en.m.wikipedia.org/wiki/PIC_microcontrollers. [Accessed 13 january 2019].
- [11] "Crystal Oscillator Circuit and Working," [Online]. Available: <https://www.elprocus.com/crystal-oscillator-circuit-and-working/>. [Accessed 1 january 2019].
- [12] "How to Program a PIC Microcontroller to Build a Project," [Online]. Available: <https://www.elprocus.com/pic-microcontroller-programming-using-c-language/>. [Accessed 1 january 2019].

Appendices

Count.asm corrections

MPASM 5.51 COUNT CORRECTIONS.ASM 1-31-
 2019 9:53:36 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
 VALUE

```

00001 ; FILE:          count.asm
00002 ; AUTHOR:        JMY
00003 ; DATE:          24th March 2005
00004 ; DESCRIPTION:   Count to 255 at 1
second intervals and display on a seven
00005 ;                segment display in
binary.
00006 ;                The pause subroutine
takes apx. 0.2 sec.
00007 ;                Pause is called 5
times
00008
Warning[215]: Processor superseded by command line.
Verify processor symbol.
00009    list   p=16F84        ; select
processor type
00010    radix   hex            ; define radix
as hexadecimal
00011 ;    #include "p16f84.inc"
00012 ;-----
-----
00013 ;    cpu equates (memory map)
00000006    00014 portB equ   0x06        ;port B
address
00000006    00015 trisb equ 0x06        ;trisB
address in register bank 1
00016

```

```

00000003      00017 status equ 0x03      ;status
register lower case s no capital
00000005      00018 RP0      equ 5      ;bit 5
of status register
00000006      00019 RP1      equ 6      ;bit 6
of status register
00020
0000000C      00021 count equ 0x0c      ; counter
0 to 255
0000000D      00022 nCount equ 0x0d      ; pause
routine inner loop counter
0000000E      00023 mCount equ 0x0e      ; pause
routine outer loop counter
00024
00025 ;-----
-----
00026
000000      00027      org 0x000
00028
000000      00029 start
000000 9A03      00030      bcf      status,RP0      ;
clear bank selection bits in the status reg
000002 9C03      00031      bcf      status,RP1
00032
000004 8A03      00033      bsf      status,RP0      ;
select bank 1
000006 0E00      00034      movlw 0x00      ; load w
with 0x00
000008 6E06      00035      movwf trisb      ; load it
into trisb to make portb all outputs
00000A 9A03      00036      bcf      status,RP0      ;
reselect bank 0
00037
00000C 6A06      00038      clrf portB      ; clear all
lines low
00000E 6A0C      00039      clrf count      ; initialize
counter to 0

```

```

00040
000010      00041 get_count
000010 500C      00042      movf  count, w      ; move
count to W
00043
00044
000012 6E06      00045      movwf portB      ; move
W to port B to update the display
00046
000014 0F01      00047      addlw  1          ; add
+1 to W
00048
000016 6E0C      00049      movwf  count      ; save W
in count
00050
000018 EC18 F000  00051      call  pause      ; delay
200ms by subroutine
00001C EC18 F000  00052      call  pause
```

MPASM 5.51 COUNT CORRECTIONS.ASM 1-31-
 2019 9:53:36 PAGE 2

LOC OBJECT CODE LINE SOURCE TEXT
 VALUE

```

000020 EC18 F000    00053    call   pause
000024 EC18 F000    00054    call   pause
000028 EC18 F000    00055    call   pause        ; five
pause executions equals ~ 1 second
                    00056
00002C EF08 F000    00057    goto   get_count        ;
repeat forever
                    00058
                    00059 ;-----
-----
                    00060 ;subroutine pause: lets waste some time....
                    00061 ;
                    00062 ;inputs: nothing
                    00063 ;outputs: nothing
                    00064 ;
                    00065 ;does: nothing but waste some time
                    00066 ;corrupts: register W
                    00067 ;
                    00068 ;description: two loops using "mCount"
and "nCount" register files
                    00069 ;                    "mCount" is the outer
loop counter (255 down to zero)
                    00070 ;                    "nCount" is the inner
loop counter (255 down to zero)
                    00071 ;
                    00072 ;-----
-----
000030              00073 pause
000030 0EFF        00074    movlw   0xff        ; set w =
255 decimal to f no three

```



```

000032 6E0E      00075      movwf  mCount      ;
mCount = w: outer loop counter count not cont
000034           00076 loadN
000034 0EFF      00077      movlw  0xff      ; set w =
255 decimal
000036 6E0D      00078      movwf  nCount      ;
nCount = w: inner loop counter
000038           00079 decN
000038 2E0D      00080      decfsz nCount, f      ;
nCount--
00003A EF1C F000  00081      goto  decN      ; if
nCount != 0 then repeat nCount--
000082
00003E 2E0E      00083      decfsz mCount, f      ; else
decrement Count
000040 EF1A F000  00084      goto  loadN      ; if
mCount != 0 then
000085                                     ; reload
nCount to 255 and decrement
000044 0012      00086      return      ; else
exit subroutine
000087
000088
000089      end; no space between en and d

```

MPASM 5.51 COUNT CORRECTIONS.ASM 1-31-
 2019 9:53:36 PAGE 3

SYMBOL TABLE

LABEL	VALUE
RP0	00000005
RP1	00000006
__18F452	00000001
count	0000000C
decN	00000038
get_count	00000010
loadN	00000034
mCount	0000000E
nCount	0000000D
pause	00000030
portB	00000006
start	00000000
status	00000003
trisb	00000006

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
0040 : XXXXXX-----
-----

```

All other memory blocks unused.

Program Memory Bytes Used: 70
 Program Memory Bytes Free: 32698

Morgan jones (1706863)



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Errors : 0

Warnings : 1 reported, 0 suppressed

Messages : 0 reported, 0 suppressed

Modify "Count.asm" so the seven segment display shows "H"
 "E" "L" "P" at a rate that is easy to read.

MPASM 5.51

HELP.ASM 1-8-2019 17:10:20 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
 VALUE

```

00001 ; FILE:          count.asm
00002 ; AUTHOR:        JMY
00003 ; DATE:          24th March 2005
00004 ; DESCRIPTION:   Count to 255 at 1 second intervals and display on a seven
00005 ;                  segment display in binary.
00006 ;                  The pause subroutine takes apx. 0.2 sec.
00007 ;                  Pause is called 5 times
00008
00009      list p=16F84      ; select processor type
00010      radix hex         ; define radix as hexadecimal
00011 ; #include "p16f84.inc"
00012 ;-----
00013 ; cpu equates (memory map)
00000006      00014 portB equ 0x06      ;port B address
00000006      00015 trisb equ 0x06      ;trisB address in register bank 1
00016
00000003      00017 Status equ 0x03      ;status register
00000005      00018 RP0 equ 5             ;bit 5 of status register
00000006      00019 RP1 equ 6             ;bit 6 of status register
00020
00021
0000000D      00022 nCount equ 0x0d      ; pause routine inner loop counter
0000000E      00023 mCount equ 0x0e      ; pause routine outer loop counter
00024
00025 ;-----
00026
0000      00027      org 0x000
00028
0000      00029 start
0000 1283      00030      bcf      Status,RP0      ; clear bank selection bits in the status reg
0001 1303      00031      bcf      Status,RP1
00032
0002 1683      00033      bsf      Status,RP0      ; select bank 1
0003 3000      00034      movlw 0x00      ; load w with 0x00
0004 0086      00035      movwf trisb      ; load it into trisb to make portb all outputs
0005 1283      00036      bcf      Status,RP0      ; reselect bank 0
00037
0006 0186      00038      clrf portB      ; clear all lines low
00039
0007      00040 write_stop
00041
0007 3009      00042      movlw 0x09 ;load LED states for letter 'H' into W reg
0008 0086      00043      movwf portB ; put W reg onto Port B
0009 2020      00044      call delay1s
00045
000A 3006      00046      movlw 0x06 ;load LED states for letter 'e' into W reg
000B 0086      00047      movwf portB
000C 2020      00048      call delay1s

```

Morgan jones (1706863)



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

```
00049
000D 3047      00050   movlw 0x47 ;load LED states for letter 'I' into W reg
000E 0086      00051   movwf portB
000F 2020      00052   call delay1s
00053
```



```

LOC OBJECT CODE   LINE SOURCE TEXT
VALUE

0010 300C      00054  movlw 0x0c ;load LED states for letter 'P' into W reg
0011 0086      00055  movwf portB
0012 2020      00056  call delay1s
                        00057
0013 307F      00058  movlw 0x7f ;load LED states for letter ' ' into W reg
0014 0086      00059  movwf portB
0015 2020      00060  call delay1s
                        00061
                        00062
0016 2807      00063  goto  write_stop      ; repeat forever
                        00064
                        00065 ;-----
00066 ;subroutine pause: lets waste some time....
00067 ;
00068 ;inputs: nothing
00069 ;outputs: nothing
00070 ;
00071 ;does: nothing but waste some time
00072 ;corrupts: register W
00073 ;
00074 ;description: two loops using "mCount" and "nCount" register files
00075 ;                "mCount" is the outer loop counter (255 down to zero)
00076 ;                "nCount" is the inner loop counter (255 down to zero)
00077 ;
00078 ;-----
0017      00079 pause
0017 30FF      00080  movlw  0xff ; set w = 255 decimal
0018 008E      00081  movwf  mCount ;mcount = w outer loop counter
0019      00082 loadN
0019 30FF      00083  movlw  0xff ; set w = inner loop counter
001A 008D      00084  movwf  nCount ; nCount != 0 then
001B      00085 decN
                        00086
001B 0B8D      00087  decfsz  nCount,f
001C 281B      00088  goto   decN
                        00089
001D 0B8E      00090  decfsz  mCount,f
001E 2819      00091  goto   loadN
                        00092
001F 0008      00093  return
                        00094
                        00095
0020      00096 delay1s
                        00097
0020 2017      00098  call  pause ; delay 200ms by subroutine
0021 2017      00099  call  pause
0022 2017      00100  call  pause
0023 2017      00101  call  pause
0024 2017      00102  call  pause; five pause executions equals ~ 1 second
                        00103
0025 0008      00104  return
                        00105
                        00106

```

Morgan jones (1706863)



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

MPASM 5.51

HELP.ASM 1-8-2019 17:10:20

PAGE 3

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

00107 end



Morgan jones (1706863)

MPASM 5.51

HELP.ASM 1-8-2019 17:10:20

PAGE 4

SYMBOL TABLE

LABEL	VALUE
RP0	00000005
RP1	00000006
Status	00000003
_16F84	00000001
decN	0000001B
delay1s	00000020
loadN	00000019
mCount	0000000E
nCount	0000000D
pause	00000017
portB	00000006
start	00000000
trisb	00000006
write_stop	00000007

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXX-----

All other memory blocks unused.

Program Memory Words Used: 38

Program Memory Words Free: 986

Errors : 0

Warnings : 0 reported, 0 suppressed

Messages : 0 reported, 0 suppressed

Modify "Count.asm" so that the piezo sounder produces a
440Hz tone while the Green button is pressed.

MPASM 5.51 440 HZ.ASM 1-30-2019
15:50:53 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

00001 ; ; FILE: count.asm
00002 ; AUTHOR: JMY
00003 ; DATE: 24th March 2005
00004 ; DESCRIPTION: Count to 255 at 1
second intervals and display on a seven

```

00005 ; segment display in
binary.
00006 ; The pause subroutine
takes apx. 0.2 sec.
00007 ; Pause is called 5
times
00008 ; AUTHOR: MJGJ
00009 ; DATE: November 2018
Warning[215]: Processor superseded by command line.
Verify processor symbol.
00010 list p=16C84 ; select
processor type
00011 radix hex ; define radix
as hexadecimal
00012 ; #include "p16f84.inc"
00013
00014 ;-----
-----
00015 ; cpu equates (memory map)
00016
00017
00000005 00018 portA equ 0x05 ; port A
address
00000005 00019 trisa equ 0x05 ; trisA address
in register bank 1
00020
00000006 00021 portB equ 0x06 ;port B address
00000006 00022 trisb equ 0x06 ;trisB
address in register bank 1
00023
00000003 00024 Status equ 0x03 ;status
register
00000005 00025 RP0 equ 5 ;bit 5 of
status register
00000006 00026 RP1 equ 6 ;bit 6 of
status register
00027

```

```

00000000      00028 black equ 0      ; black button
conncts to bit 0 port A (count up)
00000001      00029 green equ 1      ; green button
connects to bit 0 port A (count down)
00030
00031
0000000D      00032 nCount equ 0x0d    ; pause
routine inner loop counter
0000000E      00033 mCount equ 0x0e    ; pause
routine outer loop counter
00034
00035
00036 ;-----
-----
00037
000000      00038      org 0x000
00039
000000      00040 start
000000 9A03      00041      bcf      Status,RP0      ;
clear bank selection bits in the status reg
000002 9C03      00042      bcf      Status,RP1
00043
000004 8A03      00044      bsf      Status,RP0      ;
select bank 1
00045
000006 0E00      00046      movlw 0x00      ; load w
with 0x00
000008 6E06      00047      movwf trisb      ; load it
into trisb to make portb all outputs
00048
00000A 0E03      00049      movlw 0x03
00000C 6E05      00050      movwf trisa      ; setting
RA0 and RA1 as inputs
00000E 9A03      00051      bcf      Status,RP0      ;
reselect bank 0
00052

```

MPASM 5.51 440 HZ.ASM 1-30-2019
 15:50:53 PAGE 2

LOC OBJECT CODE LINE SOURCE TEXT
 VALUE

```

000010 6A06      00053      clrf  portB      ; clear all
lines low
                                00054
                                00055
                                00056
000012          00057 loop
                                00058
000012 A205      00059      btfss portA,green ; is green
button pressed
000014 EC0E F000 00060      call sounder
                                00061
                                00062
000018 EF09 F000 00063      goto  loop      ; repeat
forever
Warning[207]: Found label after column 1. (sounder)
00001C          00064 sounder
00001C 8E06      00065      bsf  portB,7      ; turn on
sounder
Warning[203]: Found opcode in column 1. (call)
00001E EC15 F000 00066 call pause
000022 9E06      00067      bcf  portB,7      ; turn off
sounder
Warning[203]: Found opcode in column 1. (call)
000024 EC15 F000 00068 call pause
Warning[203]: Found opcode in column 1. (return)
000028 0012      00069 return
                                00070
                                00071 ;-----
-----
                                00072 ;subroutine pause: lets waste some time....

```

```

00073 ;
00074 ;inputs: nothing
00075 ;outputs: nothing
00076 ;
00077 ;does: nothing but waste some time
00078 ;corrupts: register W
00079 ;
00080 ;description: two loops using "mCount"
and "nCount" register files
00081 ;          "mCount" is the outer
loop counter (255 down to zero)
00082 ;          "nCount" is the inner
loop counter (255 down to zero)
00083 ;
00084 ;-----
-----
00002A          00085 pause
00002A 0E13      00086  movlw    0x13 ; set w = 19
decimal
00002C 6E0E      00087  movwf    mCount ;mcount =
w outer loop counter
00002E          00088 loadN
00002E 0EFF      00089  movlw    0xff ; set w = inner
loop counter
000030 6E0D      00090  movwf    nCount ; nCount !=
0 then
000032          00091 decN
000032          00092
000032 2E0D      00093  decfsz   nCount,f
000034 EF19 F000  00094  goto     decN
000034          00095
000038 2E0E      00096  decfsz   mCount,f
00003A EF17 F000  00097  goto     loadN
00003A          00098
00003E 0012      00099  return
00100
00101

```

MPASM 5.51 440 HZ.ASM 1-30-2019
15:50:53 PAGE 3

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```
000040          00102 delay1s
                00103
000040 EC15 F000  00104  call pause ; delay 200ms by
subroutine
000044 EC15 F000  00105  call pause
000048 EC15 F000  00106  call pause
00004C EC15 F000  00107  call pause
000050 EC15 F000  00108  call pause ; five pause
executions equals ~ 1 second
                00109
000054 0012      00110  return
                00111
                00112
                00113  end
```

MPASM 5.51 440 HZ.ASM 1-30-2019
 15:50:53 PAGE 4

SYMBOL TABLE

LABEL	VALUE
RP0	00000005
RP1	00000006
Status	00000003
__18F452	00000001
black	00000000
decN	00000032
delay1s	00000040
green	00000001
loadN	0000002E
loop	00000012
mCount	0000000E
nCount	0000000D
pause	0000002A
portA	00000005
portB	00000006
sounder	0000001C
start	00000000
trisa	00000005
trisb	00000006

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXXXXXX XXXXXX-----
-----

```

All other memory blocks unused.

Morgan jones (1706863)

Program Memory Bytes Used: 86
Program Memory Bytes Free: 32682

Errors : 0
Warnings : 5 reported, 0 suppressed
Messages : 0 reported, 0 suppressed

Modify "Count.asm" so that the piezo sounder produces a low frequency tone while the Green button is pressed and a high frequency tone while the Black button is pressed.

MPASM 5.51 LOW HIGH FREQUENCY.ASM 1-8-2019 17:40:54 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00001 ; ; FILE:          count.asm
00002 ; AUTHOR:         JMY
00003 ; DATE:           24th March 2005
00004 ; DESCRIPTION:    Count to 255 at 1
second intervals and display on a seven
00005 ;                segment display in
binary.
00006 ;                The pause subroutine
takes apx. 0.2 sec.
00007 ;                Pause is called 5
times
00008 ; AUTHOR:         MJGJ
00009 ; DATE:           November 2018
Warning[215]: Processor superseded by command line.
Verify processor symbol.
00010 list p=16C84      ; select
processor type
00011 radix hex        ; define radix
as hexadecimal
00012 ; #include "p16f84.inc"
00013
00014 ;-----
-----
00015 ; cpu equates (memory map)
00016
00017

```

```

00000005      00018 portA equ 0x05      ; port A
address
00000005      00019 trisa equ 0x05      ; trisA address
in register bank 1
00020
00000006      00021 portB equ 0x06      ;port B address
00000006      00022 trisb equ 0x06      ;trisB
address in register bank 1
00023
00000003      00024 Status equ 0x03      ;status
register
00000005      00025 RP0      equ 5      ;bit 5 of
status register
00000006      00026 RP1      equ 6      ;bit 6 of
status register
00027
00000000      00028 black equ 0      ; black button
connnects to bit 0 port A (count up)
00000001      00029 green equ 1      ; green button
connects to bit 0 port A (count down)
00030
00031
0000000D      00032 nCount equ 0x0d      ; pause
routine inner loop counter
0000000E      00033 mCount equ 0x0e      ; pause
routine outer loop counter
00034
00035
00036 ;-----
-----
00037
0000      00038      org 0x000
00039
0000      00040 start
0000 1283      00041      bcf      Status,RP0      ; clear
bank selection bits in the status reg
0001 1303      00042      bcf      Status,RP1

```

```
00043
0002 1683      00044      bsf      Status,RP0      ;
select bank 1
00045
0003 3000      00046      movlw 0x00      ; load w
with 0x00
0004 0086      00047      movwf trisb      ; load it
into trisb to make portb all outputs
00048
0005 3003      00049      movlw 0x03
0006 0085      00050      movwf trisa      ; setting RA0
and RA1 as inputs
0007 1283      00051      bcf      Status,RP0      ;
reselect bank 0
00052
```

MPASM 5.51 LOW HIGH FREQUENCY.ASM 1-8-
 2019 17:40:54 PAGE 2

LOC OBJECT CODE LINE SOURCE TEXT
 VALUE

```

0008 0186      00053      clrf  portB      ; clear all
lines low
                                00054
                                00055
                                00056
0009      00057 loop
0009 1C05      00058      btfss portA,black ; is black
button pressed
000A 200E      00059      call sounder
                                00060
000B 1C85      00061      btfss portA,green ; is green
button pressed
000C 200E      00062      call sounder
                                00063
                                00064
000D 2809      00065      goto  loop      ; repeat
forever
Warning[207]: Found label after column 1. (sounder)
000E      00066 sounder
000E 1786      00067      bsf  portB,7      ; turn on sounder
Warning[203]: Found opcode in column 1. (call)
000F 2013      00068 call pause
0010 1386      00069      bcf  portB,7      ; turn off sounder
Warning[203]: Found opcode in column 1. (call)
0011 2013      00070 call pause
Warning[203]: Found opcode in column 1. (return)
0012 0008      00071 return
                                00072
                                00073 ;-----
-----

```

```

00074 ;subroutine pause: lets waste some time....
00075 ;
00076 ;inputs: nothing
00077 ;outputs: nothing
00078 ;
00079 ;does: nothing but waste some time
00080 ;corrupts: register W
00081 ;
00082 ;description: two loops using "mCount"
and "nCount" register files
00083 ;                "mCount" is the outer
loop counter (255 down to zero)
00084 ;                "nCount" is the inner
loop counter (255 down to zero)
00085 ;
00086 ;-----
-----
0013                00087 pause
0013 3013          00088 movlw    0x13 ; set w = 19
decimal
0014 008E          00089 movwf    mCount ;mcount = w
outer loop counter
0015                00090 loadN
0015 30FF          00091 movlw    0xff ; set w = inner
loop counter
0016 008D          00092 movwf    nCount ; nCount != 0
then
0017                00093 decN
0017                00094
0017 0B8D          00095 decfsz    nCount,f
0018 2817          00096 goto     decN
0018                00097
0019 0B8E          00098 decfsz    mCount,f
001A 2815          00099 goto     loadN
001A                00100
001B 0008          00101 return

```

MPASM 5.51 LOW HIGH FREQUENCY.ASM 1-8-
2019 17:40:54 PAGE 3

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00102
00103
001C      00104 delay1s
00105
001C 2013      00106  call pause ; delay 200ms by
subroutine
001D 2013      00107  call pause
001E 2013      00108  call pause
001F 2013      00109  call pause
0020 2013      00110  call pause ; five pause executions
equals ~ 1 second
00111
0021 0008      00112  return
00113
00114
00115      end
```

MPASM 5.51 LOW HIGH FREQUENCY.ASM 1-8-
 2019 17:40:54 PAGE 4

SYMBOL TABLE

LABEL	VALUE
RP0	00000005
RP1	00000006
Status	00000003
__16F84	00000001
black	00000000
decN	00000017
delay1s	0000001C
green	00000001
loadN	00000015
loop	00000009
mCount	0000000E
nCount	0000000D
pause	00000013
portA	00000005
portB	00000006
sounder	0000000E
start	00000000
trisa	00000005
trisb	00000006

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : XXXXXXXXXXXXXXXXXXXX
 XXXXXXXXXXXXXXXXXXXX XX-----

All other memory blocks unused.

Program Memory Words Used: 34
 Program Memory Words Free: 990

Morgan jones (1706863)

Errors : 0
Warnings : 5 reported, 0 suppressed
Messages : 0 reported, 0 suppressed

Ass 1 final

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9



Comment 1

... has a set frequency?

PAGE 10

PAGE 11

PAGE 12

PAGE 13



Comment 2

There is a crystal oscillator which uses the resonance of a vibrating crystal of piezoelectric material.

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19



Comment 3

Your flowchart should include a test to see if the green button has been pressed. If it has been pressed then the program should make the PIC produce a single cycle of a 440 Hz square wave and return to the start. If the button has not been pressed then the programme should make the PIC repeat the test on the button.

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

GRADING FORM: SMU GENERIC_SMOSEY

MORGAN JONES

59.80

ENGAGEMENT WITH LITERATURE

including reading, referencing, academic conventions and academic honesty



A comprehensive range of literature embedded in the work. Excellent selection of relevant sources.

14

High-level referencing skills, consistently applied

KNOWLEDGE & UNDERSTANDING

Knowledge of the basic underlying concepts and principles of a subject.



Knowledge is accurate with a good understanding of the field of study. Only a few minor errors, mostly relating to tests within flowcharts.

11

COGNITIVE & INTELLECTUAL

Evaluate underlying concepts and principles of a subject and interpret qualitative and quantitative data in order to develop lines of argument and make sound judgements.



Not in mark scheme

PRACTICAL APPLICATION SKILLS

Different approaches to solving problems in particular contexts.



A good and appropriate application of standard methods.

23.2

TRANSFERABLE SKILLS

Communicate the results of their study/work accurately and reliably, and with structured and coherent arguments; the qualities needed for employment requiring the exercise of some personal responsibility within a structured and managed environment.



Can communicate effectively in a suitable format, but may have minor errors. Mostly coherent work and is in a suitable structure. Your work is getting better. Well done

11.6