JSPM's

Rajarshi Shahu College of Engineering

(An Autonomous Institute affiliated to SPPU, Pune)

Department of Electrical Engineering

Class: T.Y. B.TECH

Subject: PIC Microcontroller and its Applications

Laboratory Manual

Vision of the Institute

To satisfy the aspirations of the youth force who want to lead the nation towards prosperity through techno-economic development.

Mission of the Institute

To provide, nurture and maintain an environment of high academic excellence, research, and entrepreneurship for all aspiring students which will prepare them to face global challenges maintaining high ethical and moral standards.

Vision of the Department

To develop globally competent Electrical Engineers by providing industry oriented academic environment that inculcates professional skills and ethics for techno-social benefits.

Mission of the Department

1. To transform students into successful professionals by inculcating comprehensive knowledge of Electrical Engineering.
2. To develop a conducive environment through creativity, innovation and industry institute interaction.
3. To encourage and enable students for higher education, research and entrepreneurship.

Program Outcomes (PO's)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Outcomes (CO's)

CO1	**Define architecture of PIC18F458 microcontroller.**
CO2	**Classify program for specific applications.**
CO3	**Develop programming of PIC microcontroller.**
CO4	**Analyze applications of PIC microcontroller.**
CO5	**Develop advance programming of PIC microcontroller.**
CO6	**Compile and interface of PIC microcontroller.**

List of Experiments

Sr. No.	Title of Experiment	CO
1	Introduction of MPLAB IDE software.	**CO 1**
2	Assembly language program for addition, subtraction and multiply of 8 bit number stored in array.	**CO 2**
3	Program on square wave generation using timer & delay program.	**CO 4**
4	Program on data transfer to ports.	**CO 5**
5	Program for PIC18F458 for LED interfacing.	**CO 5**
6	Program on PIC18F458 for DC motor interfacing.	**CO 5**
7	Program on PIC18F458 for stepper motor interfacing.	**CO 6**
8	Interface relay with PIC18F458 microcontroller.	**CO 6**
Extra Experiments		
9	Program for PIC18F458 for LCD interfacing.	**CO 6**
10.	Program on PIC18F458 for DAC 0808 interfacing.	**CO 5**

Experiment No. 1

Aim: Introduction of MPLAB IDE software.

Program statement: Identification & observation of PIC 18 system board on the kit, introductionof MPLAB IDE software.

Explanation:

Introduction: Micro-PIC18F microcontroller board has been specifically designed keeping in mind the needs of students for learning the PIC architecture. The board gives a complete overview for interfacing various peripheral devices which are used in the industry and consumer devices alike. A hands-on with the board will develop in the student the experience to design and implement various devices and products based on the PIC Microcontroller.

High-Performance RISC CPU

- Source code compatible with the PIC16 and PIC17 instruction sets
- Linear data memory addressing to 1536 bytes
- Linear program memory addressing to 2 Mbytes
- DC - 40 MHz osc./clock input
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier
- 16 kBytes Flash Program Memory
- 1536 Byte RAM Data Memory
- 256 Byte RAM Data Memory

Peripheral Features

- High current sink/source 25mA/25mA
- Two 16-bit timer/counter (TMR1, TMR3)
- One 8-bit/16-bit timer/counter with prescaler
- One 8-bit timer/counter with 8-bit period register
- Capture 16-bit, max. resolution 6.25ns(TCY/16)
- Compare 16-bit, max. resolution 100ns
- 3-wire SPI with Interrupt-on-Address Bit
- I²C Master and Slave mode
- Secondary Oscillator Clock Option - Timer1/Timer3

- 1,2 or 4 PWM Outputs with Selectable Polarity
- Eight Channel 10-bit Analog-to-Digital Converter

Advanced CAN Technology

- Complies with ISO CAN Conformance Test
- Message bit rates up to 1 Mbps
- 8-byte Message Length
- 29-bit Identifier Fields
- 3 Transmit Message Buffers with Prioritization
- 6 full, 29-bit Acceptance Filters
- Advanced Error Management Features

Special Microcontroller Features

- Power-On Reset
- Power-up Timer (PWRT) and Oscillator Start-Up Timer (OST)
- 1,000 erase/write cycles Enhanced Flash Program Memory
- 1,000,000 typical erase/write cycles EEPROM Data Memory
- Watchdog Timer (WDT) with its own On-Chip RC oscillator
- Programmable Code Protection
- Power Saving SLEEP mode
- 4X Phase Lock Loop (of primary oscillator)
- Secondary Oscillator (32kHz) clock input
- In-Circuit Debug (ICD)

CMOS Technology

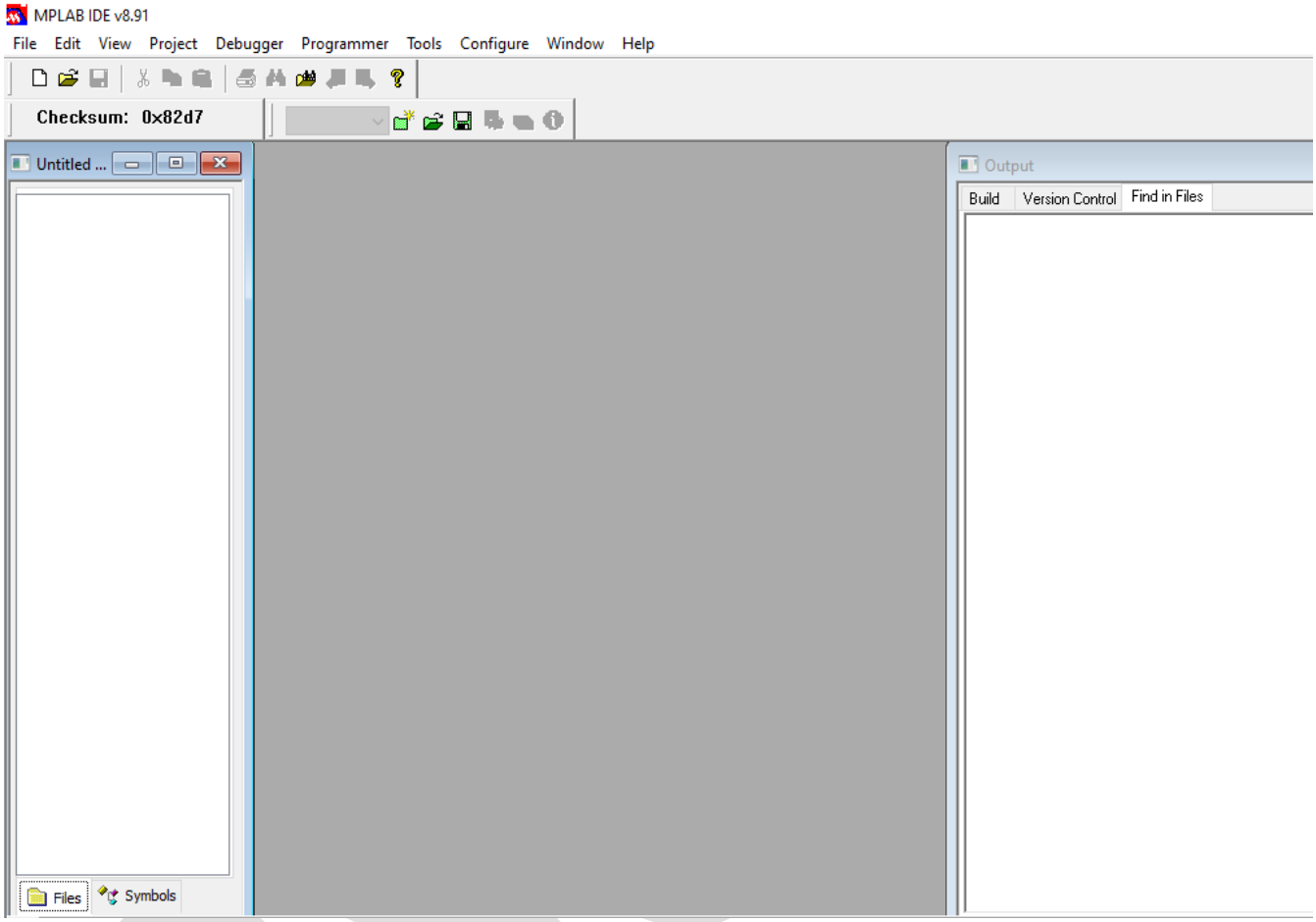
- Low power, high speed CMOS FLASH technology
- Fully Static Design
- Wide Operating Voltage Range (2.0V to 5.5V)

I/O and Packages

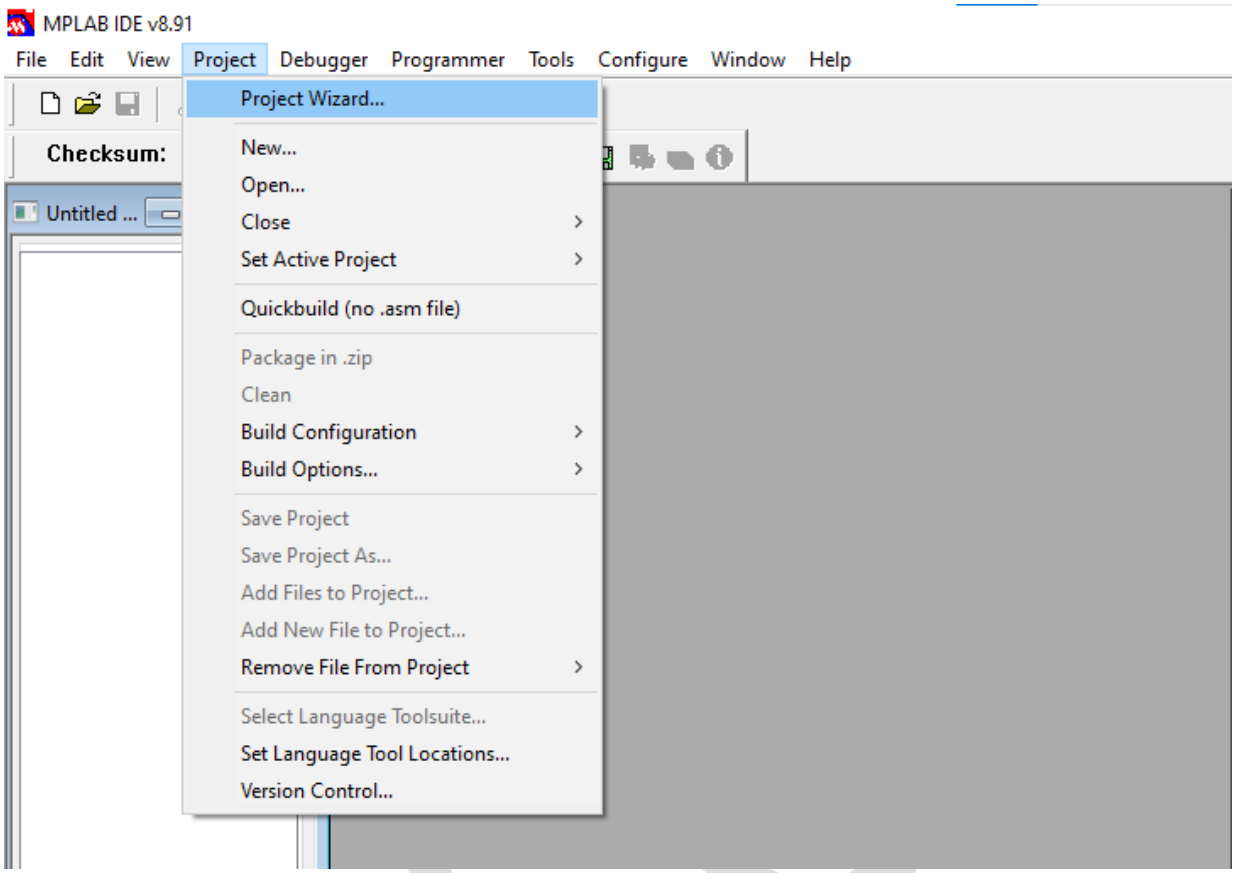
- 33 I/O pins with individual direction control
- 40-pin DIP

Creating a New Project using MPLAB IDE:

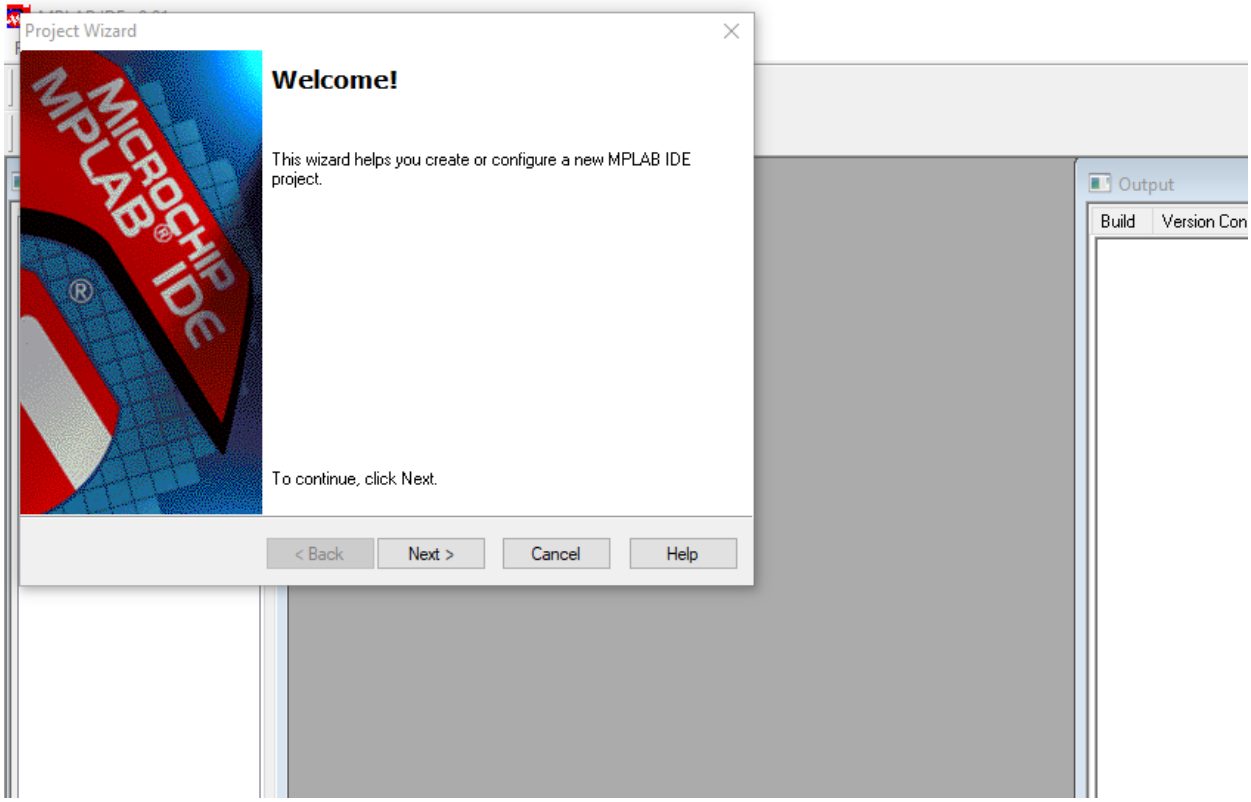
Step 1: Launch the MPLAB IDE.



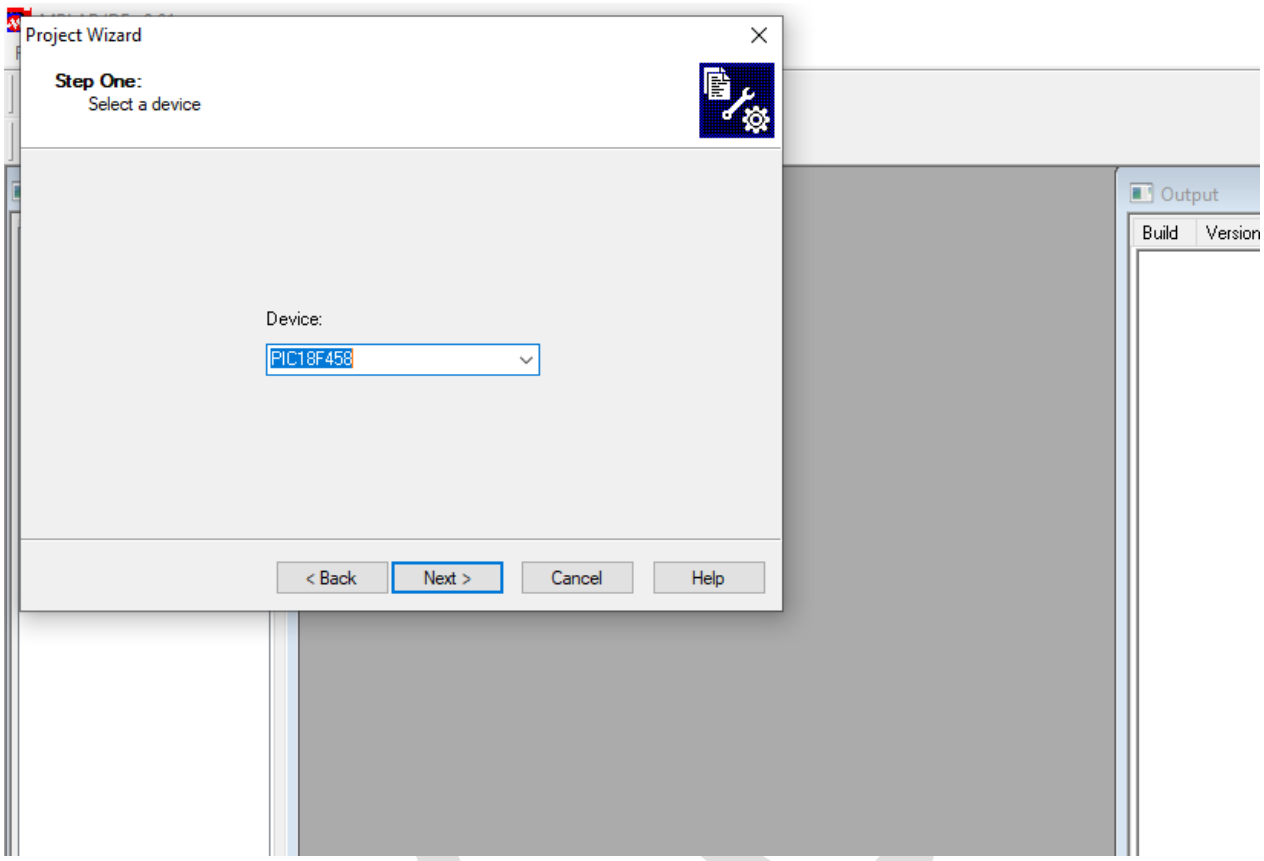
Step 2: Click on Project → select Project wizard



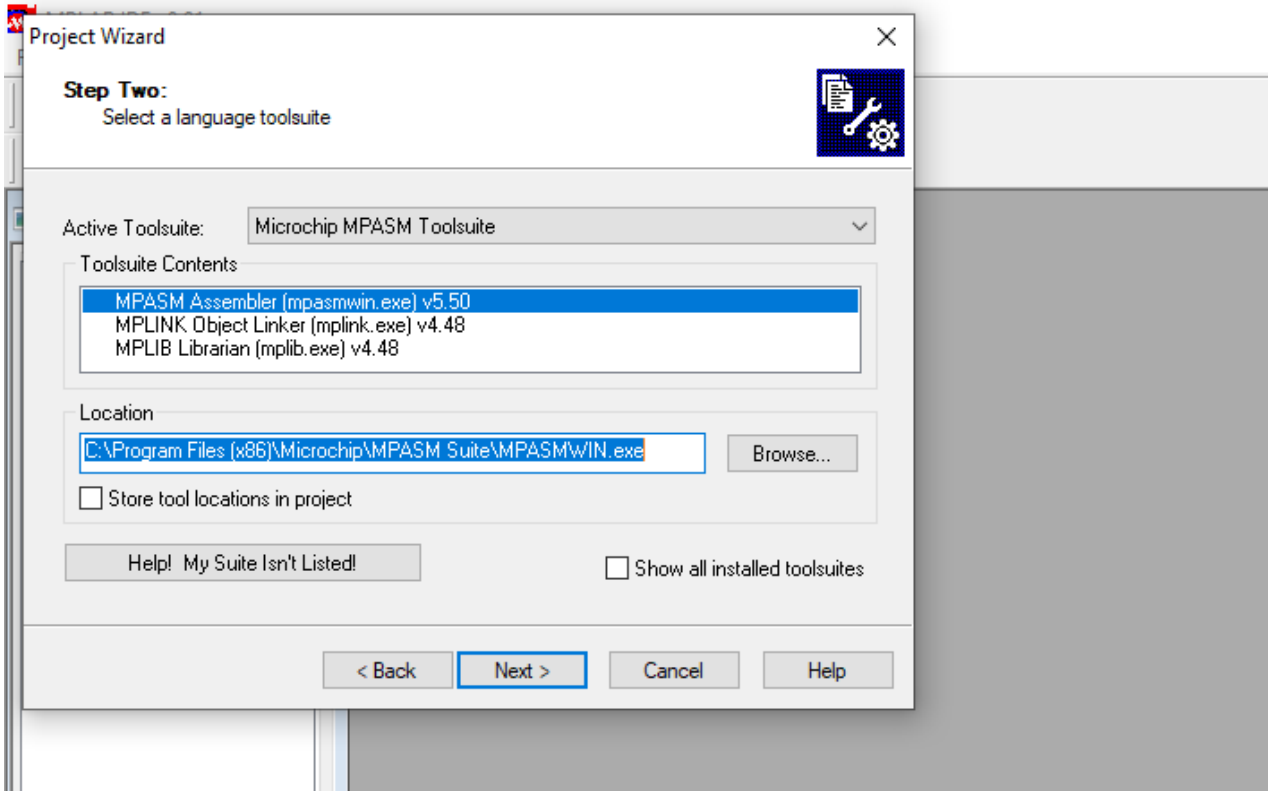
Step 3: Select → Next



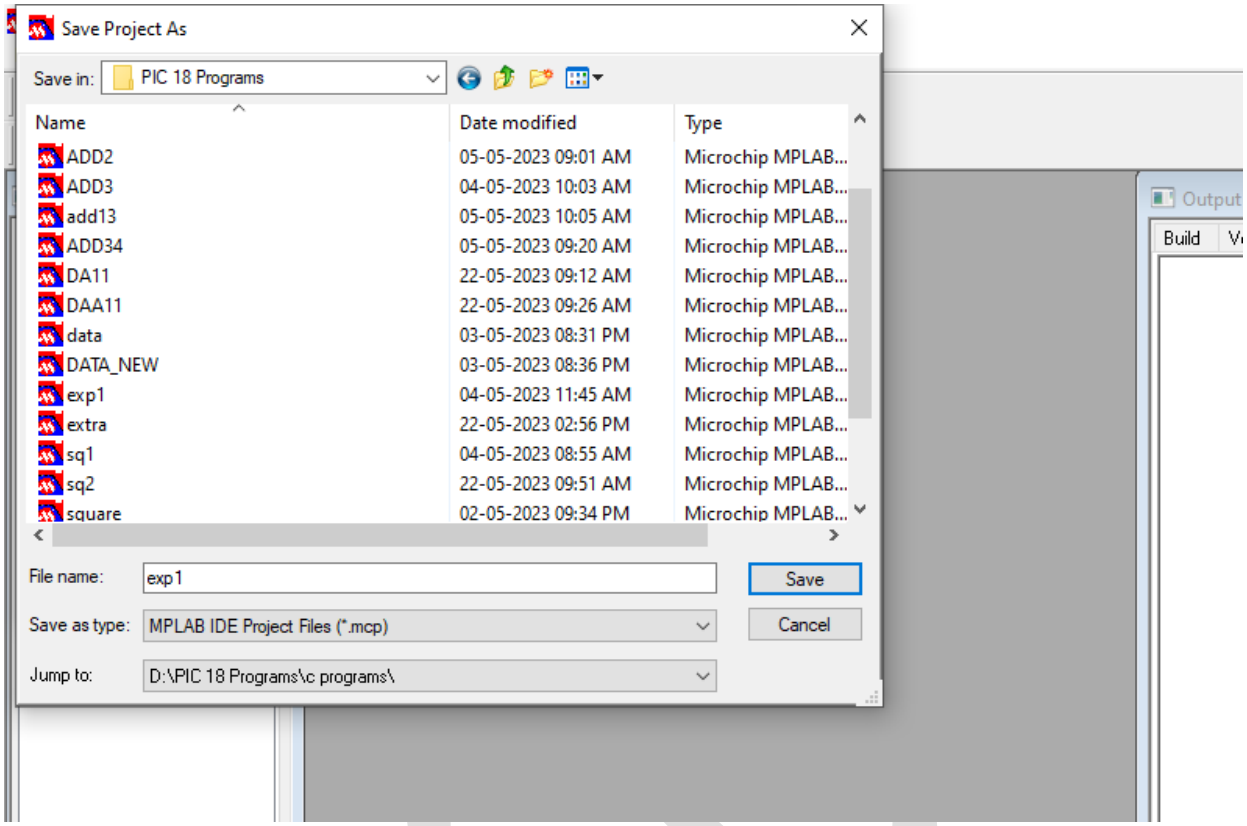
Step 4: Select → PIC18F458 Microcontroller



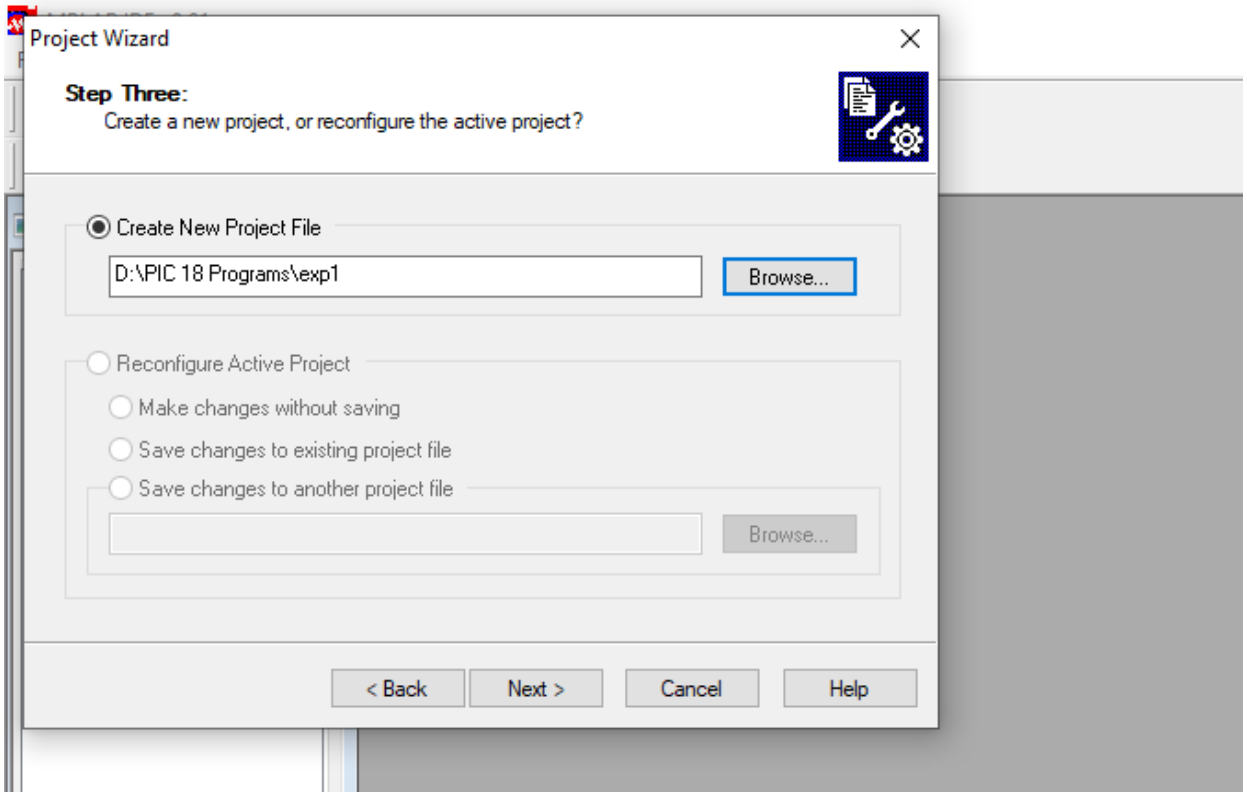
Step 5: Select → microchip MPASM Toolsuite then select → MPASM Assembler



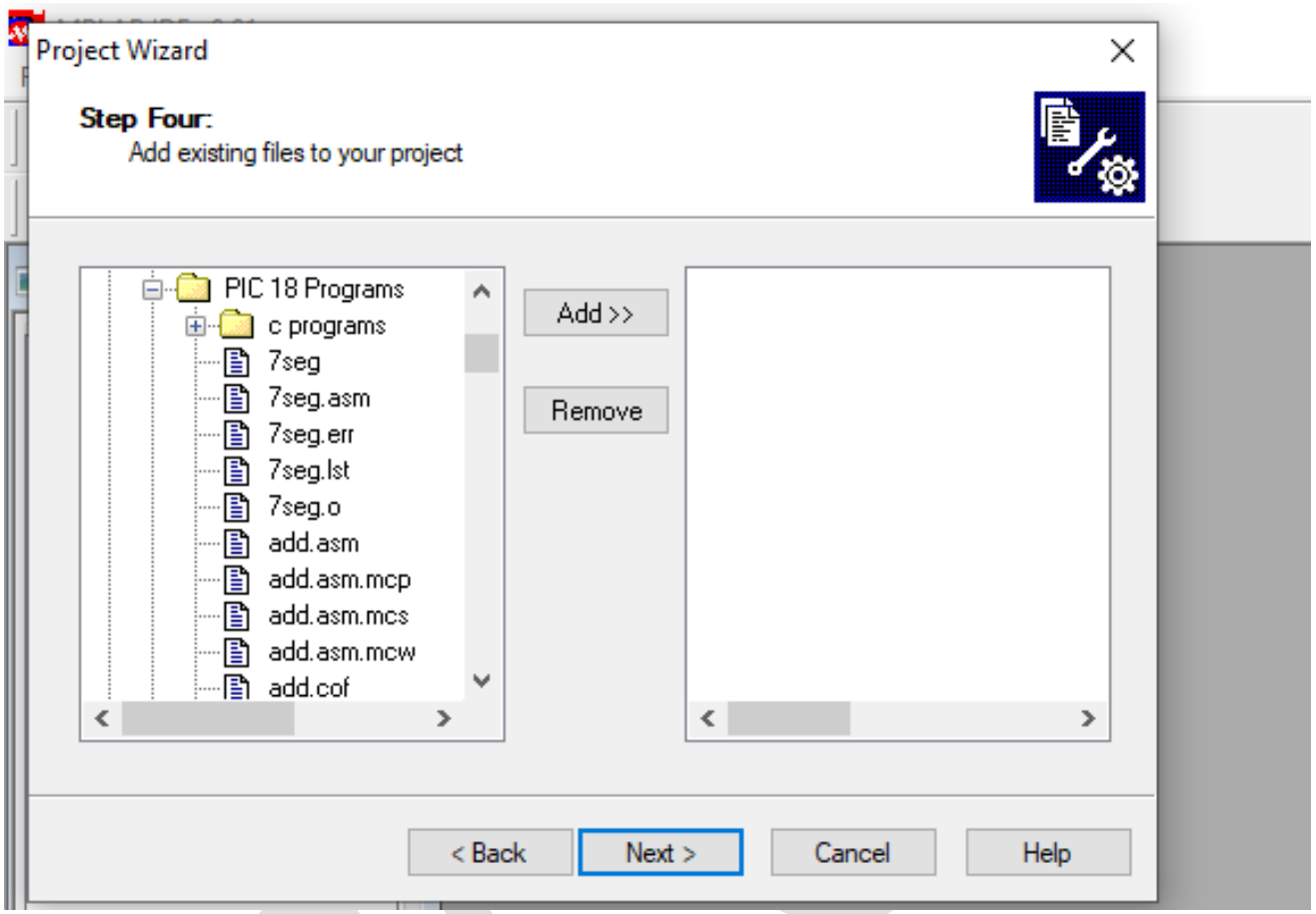
Step 6: Create folder then save file in the same folder with exp1 name.



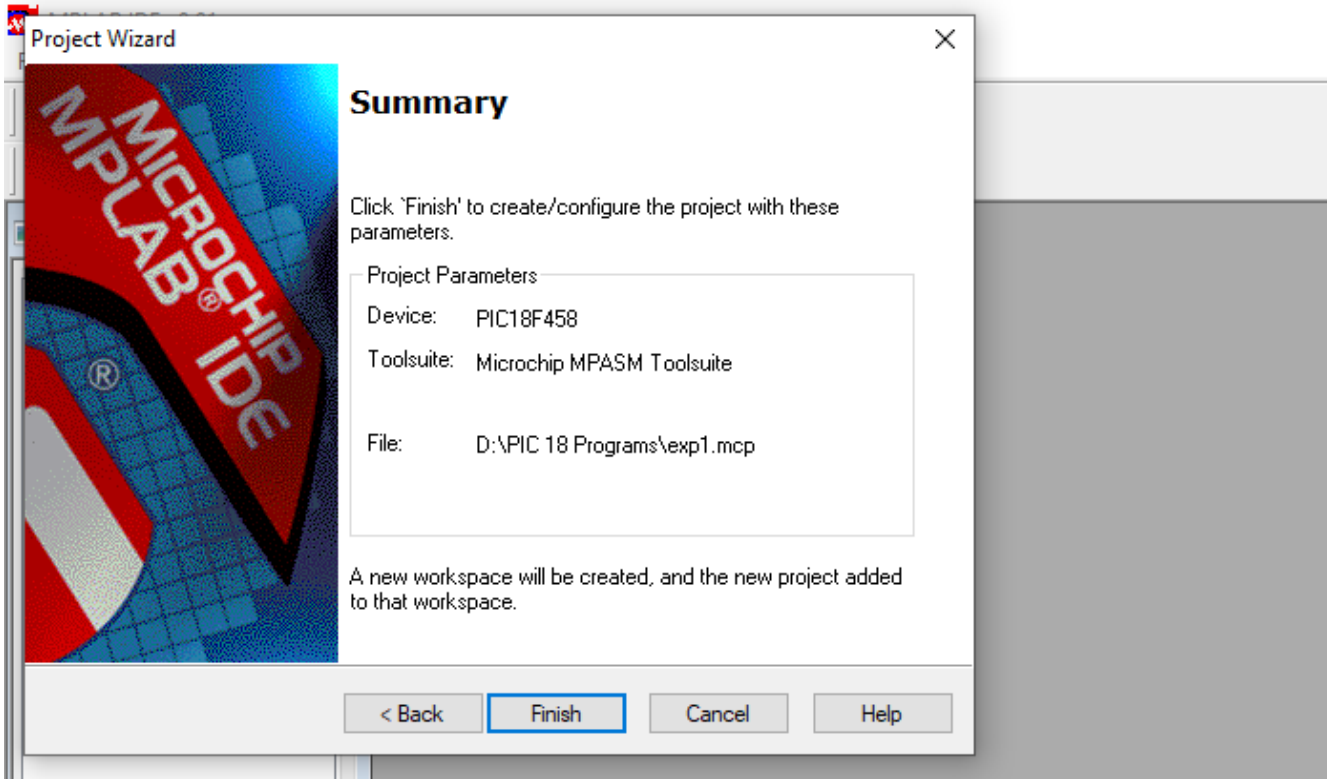
Step 7: Then browse same folder name here.



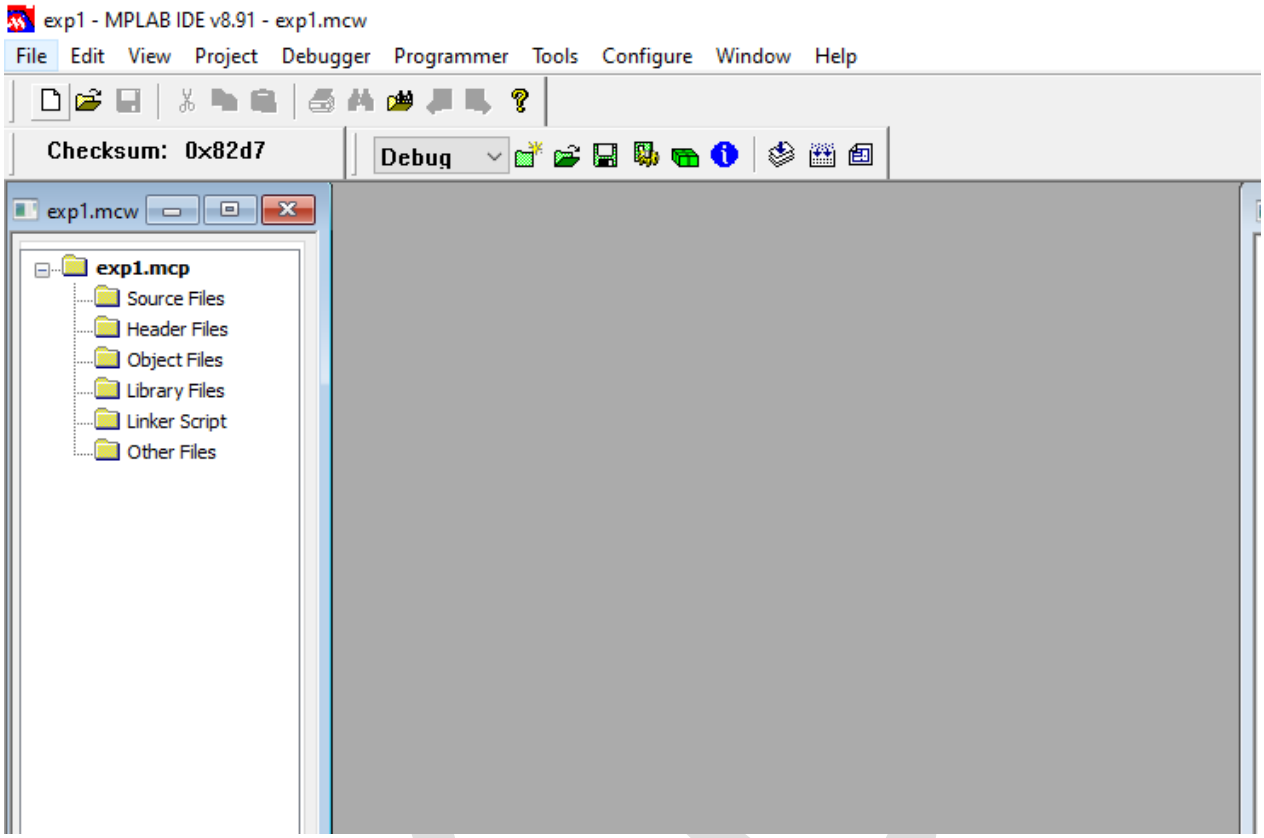
Step 8: Select → next



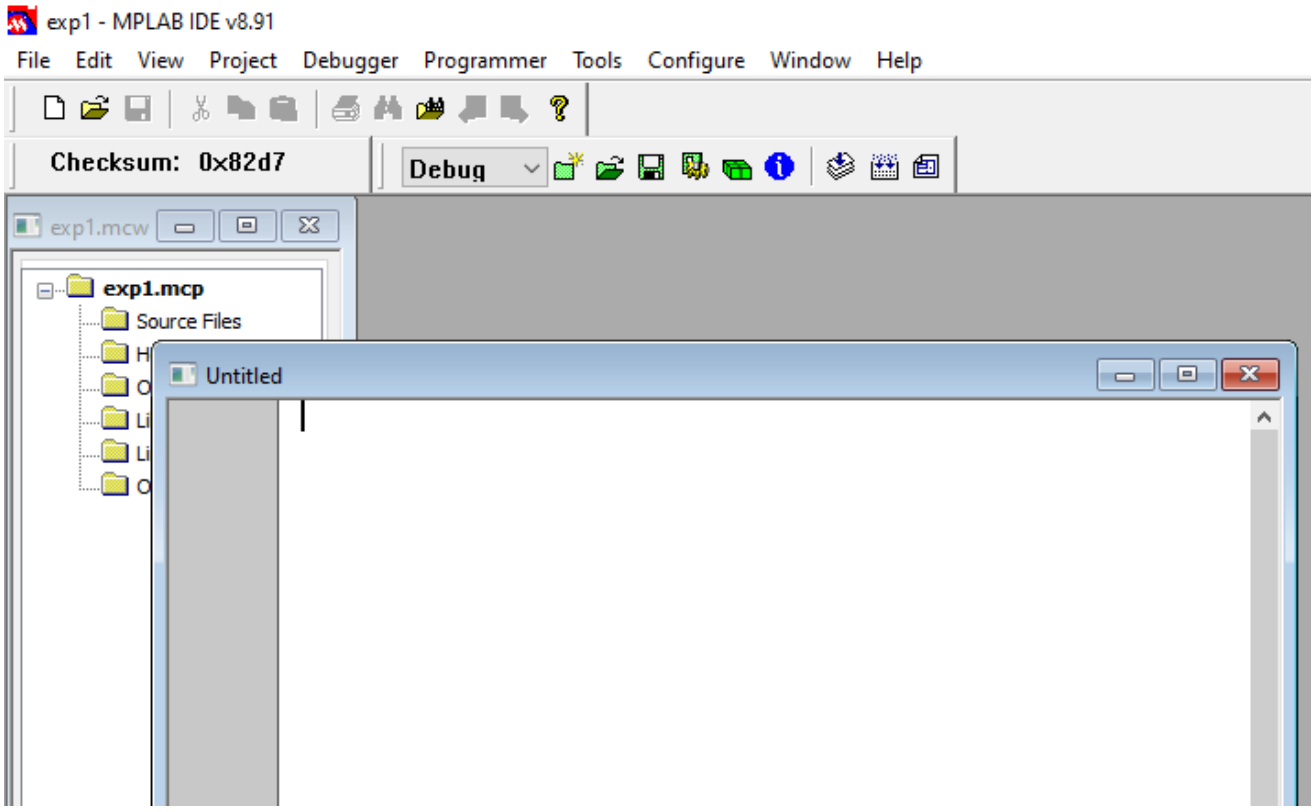
Step 9: Select → Finish

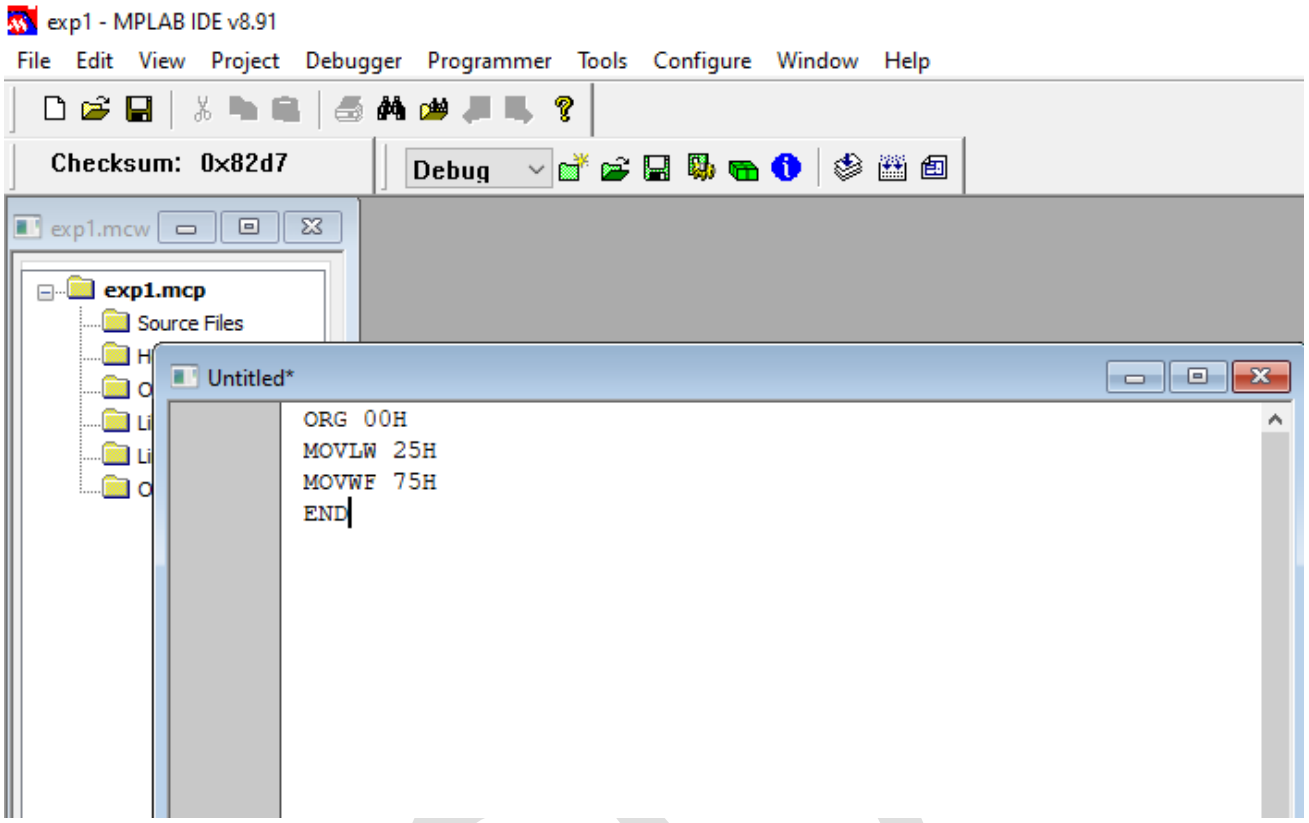


Step 10: Move to tab file and create new file.

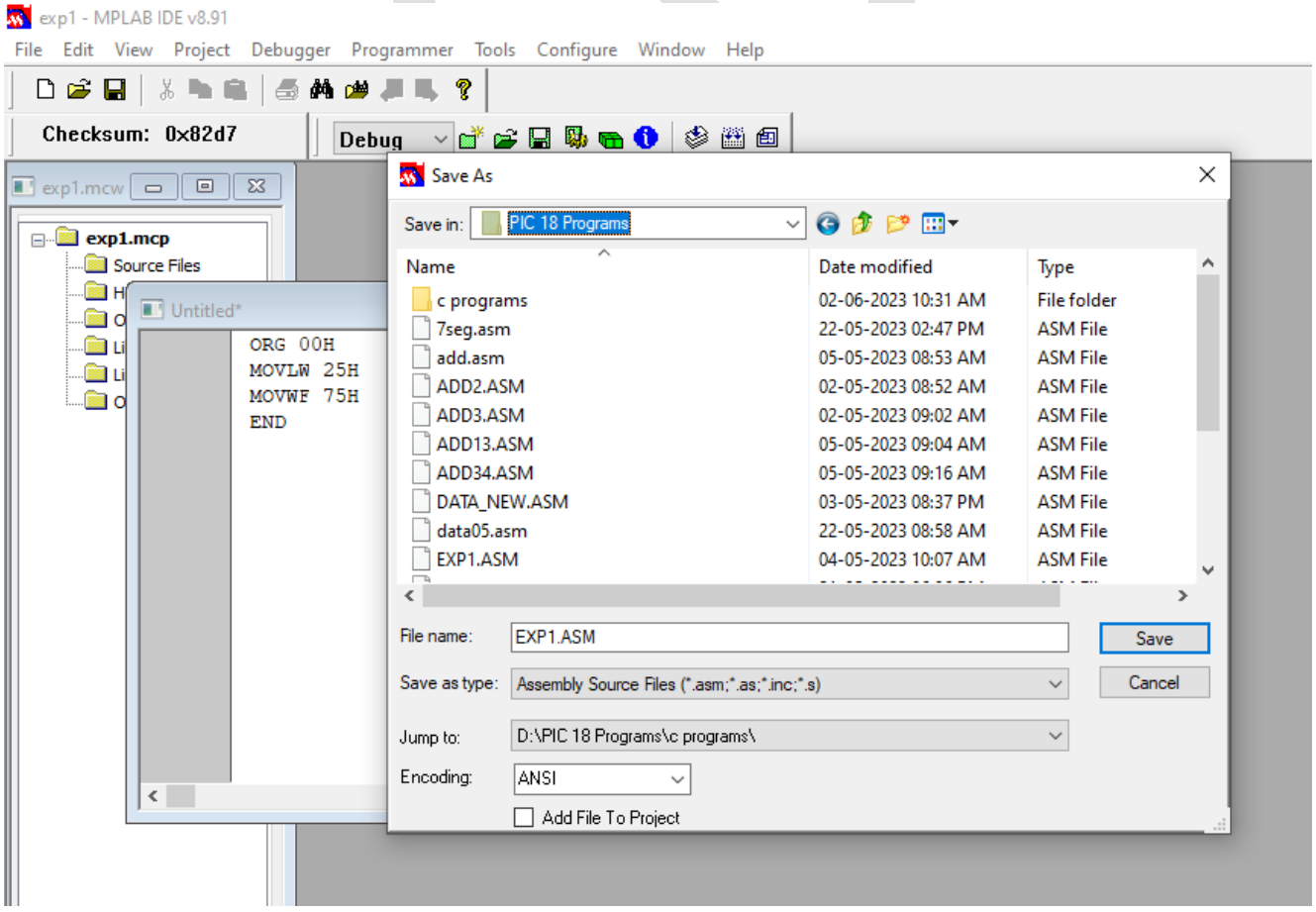


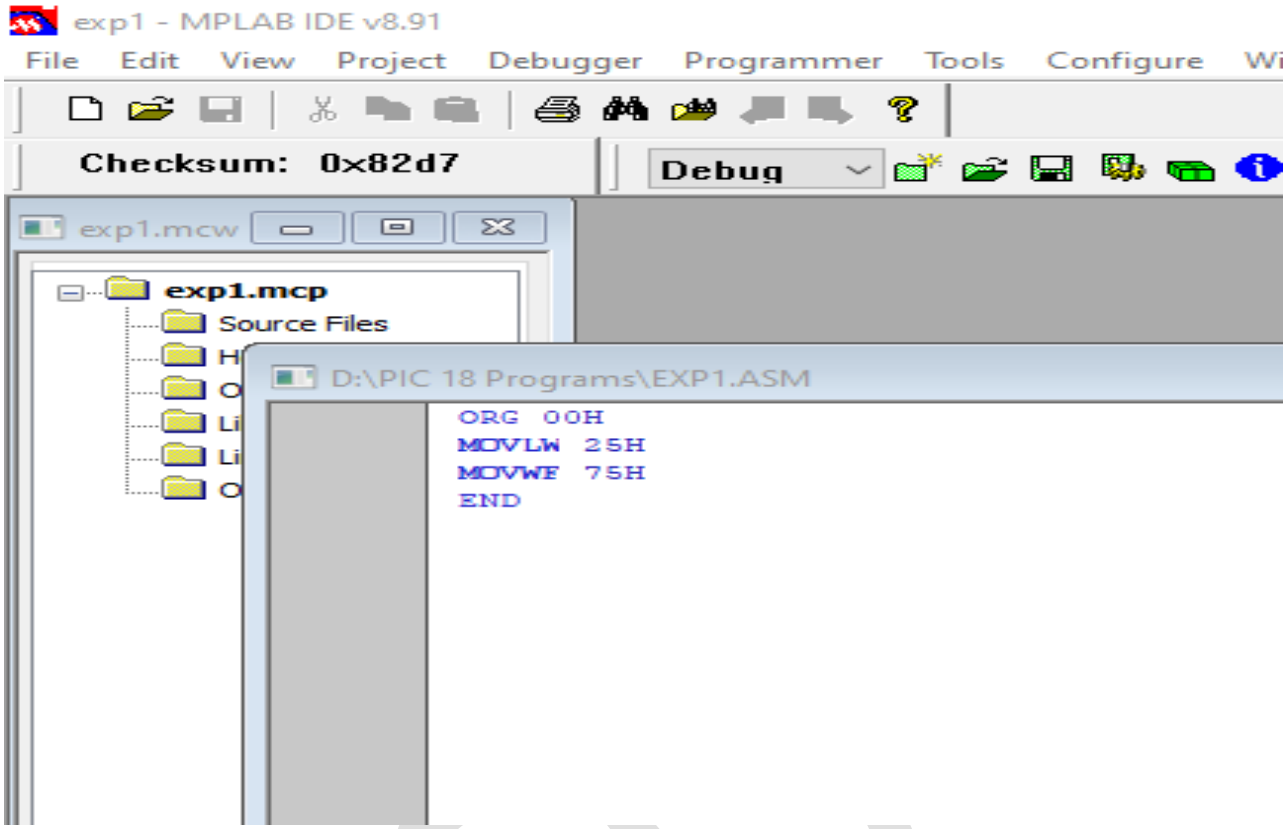
Step 11: Type program in untitled file.



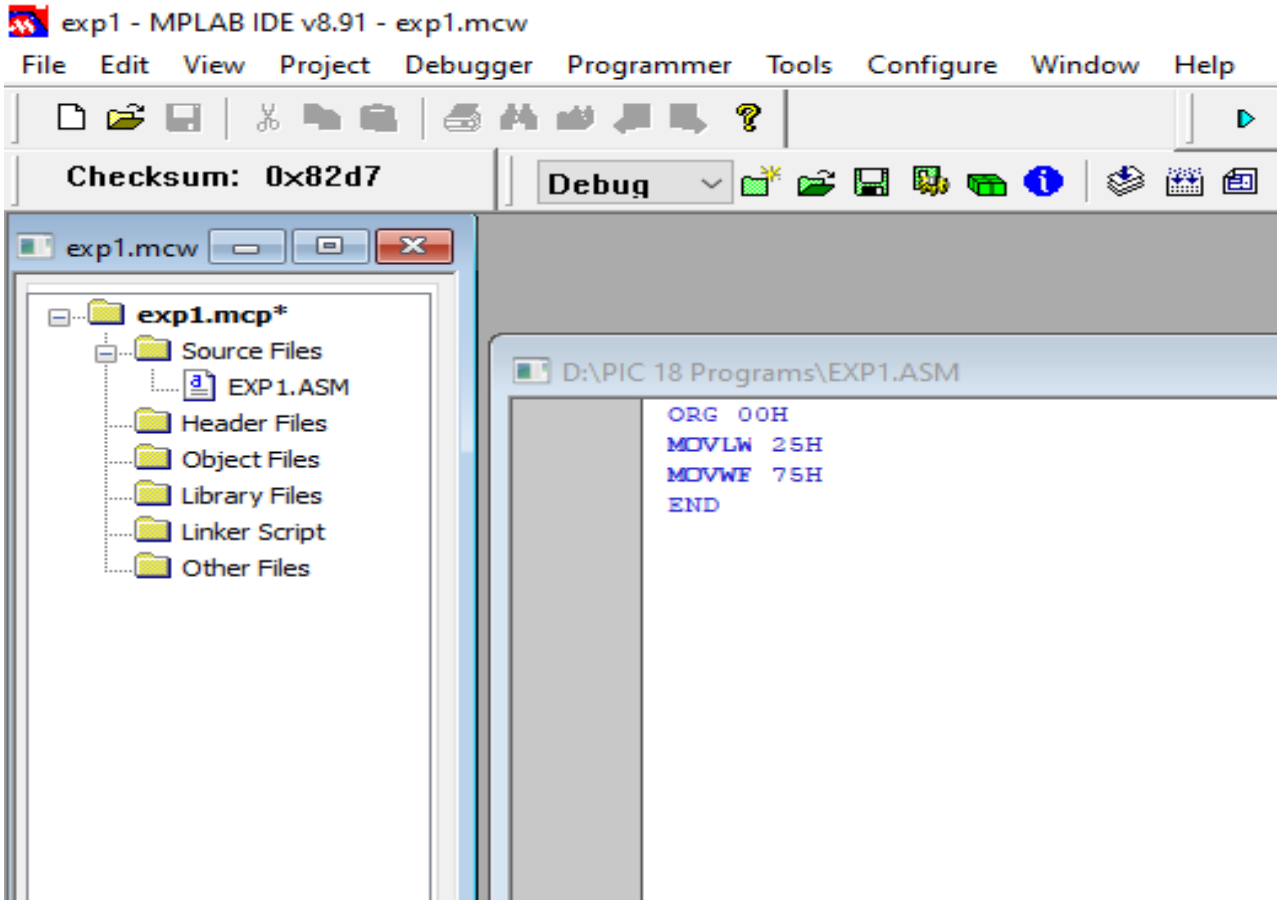


Step 12: Select file and save file with name exp1.asm

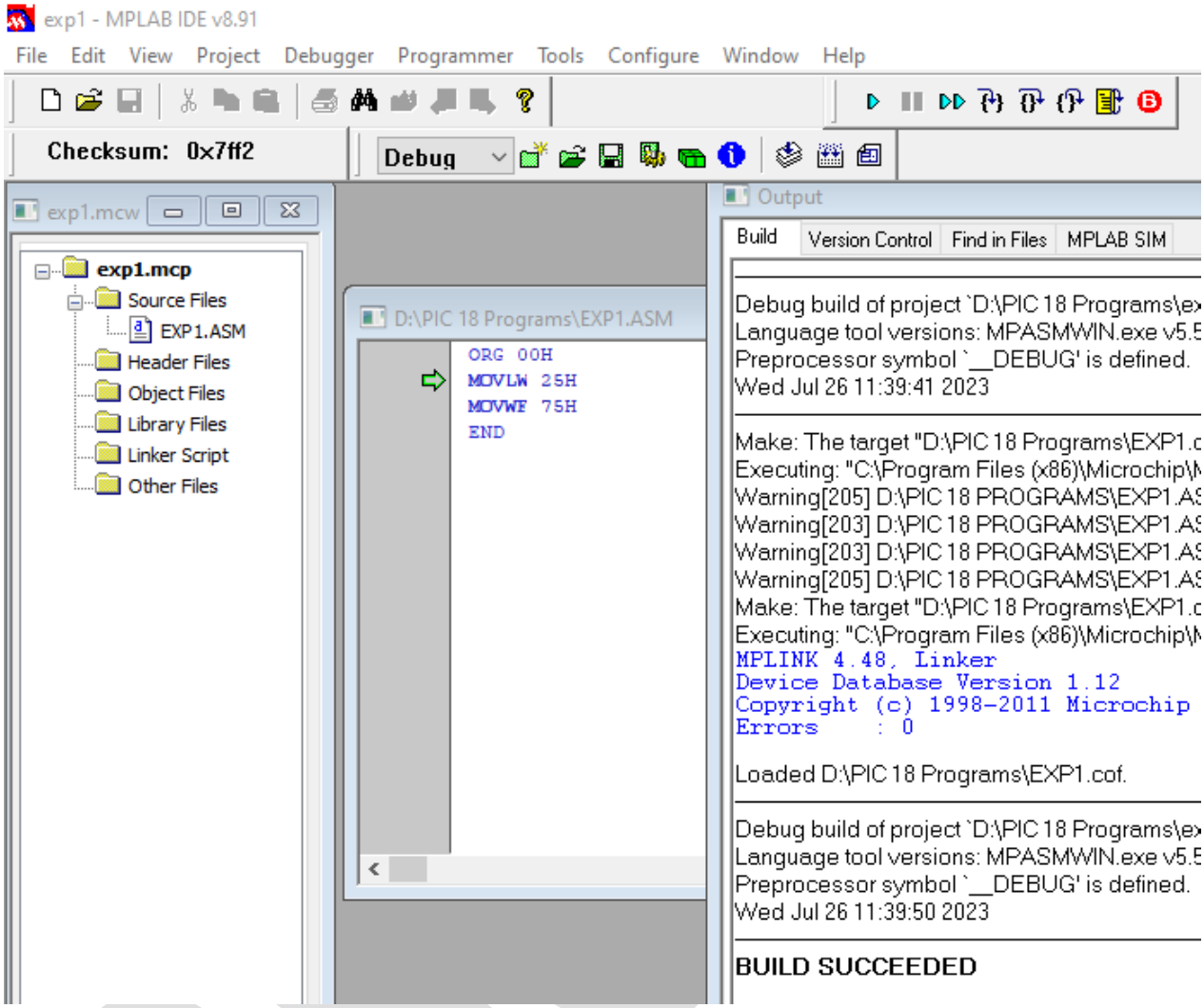




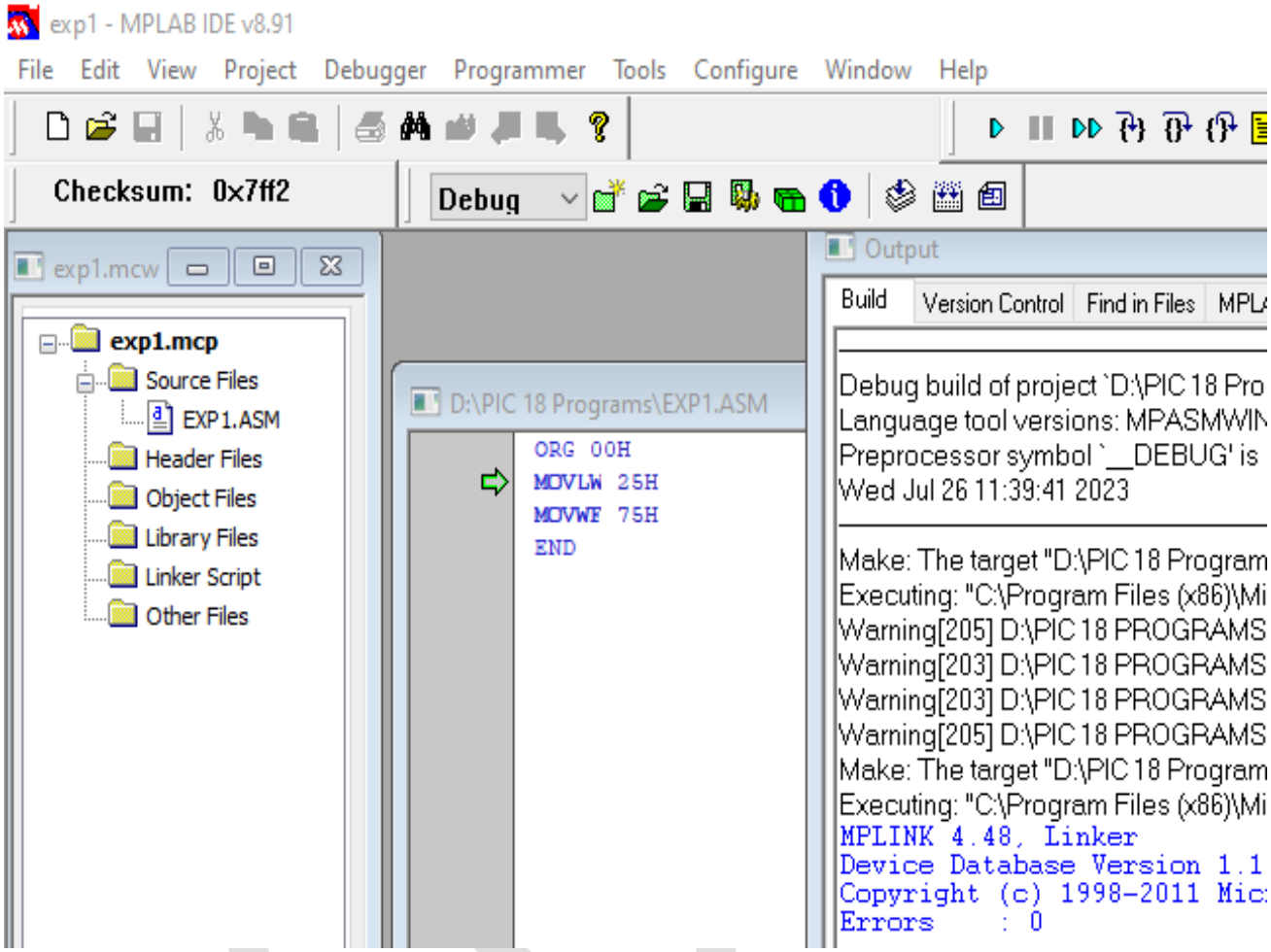
Step 13: Add exp1.asm file in source files.



Step 14: Then select Project → Project Wizard → BUILD ALL

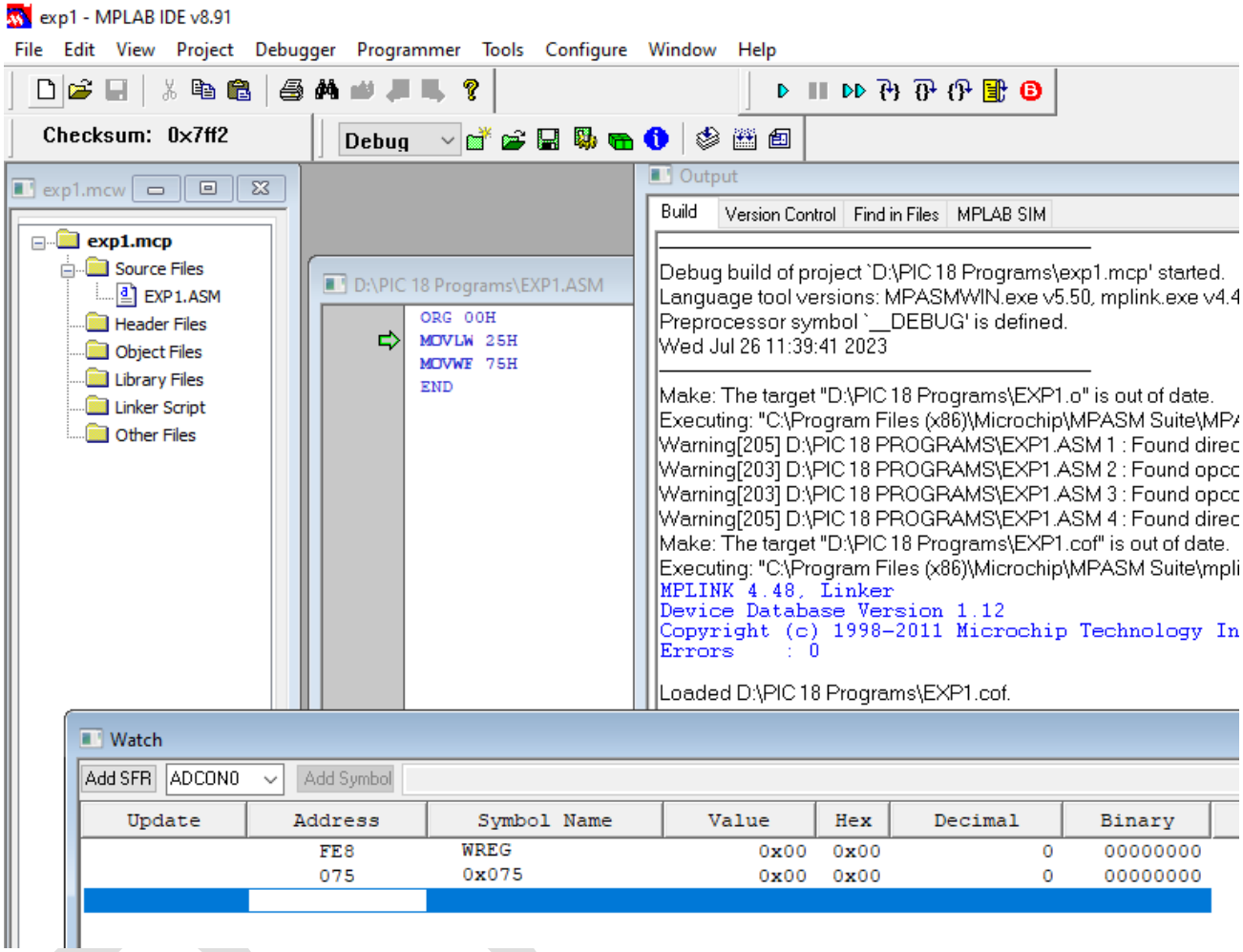


Step 15: Mote to tab debugger the select the MPLAB Simulator



Step 16: Then select the green button for run the program.

To check output move to tab view →watch → tab symbol → Symbol Name WREG → address 075H



Conclusion: Using assembly language programming date is transfer value from WREG to any address.

Experiment No. 2(A)

Aim: Assembly language program for addition of 8 bit number stored in array.

Apparatus: MPLAB IDE software

Program Statement:

Write ALP in PIC18F458 for addition two 8 bit numbers. Explanation: There are two 8 bit numbers. Clear Working register. Load first number to Working register and second number to B register. Use ADDLW 75H instruction for addition. ADD adds the byte variable indicated to the Working register, leaving the result in the Working register. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

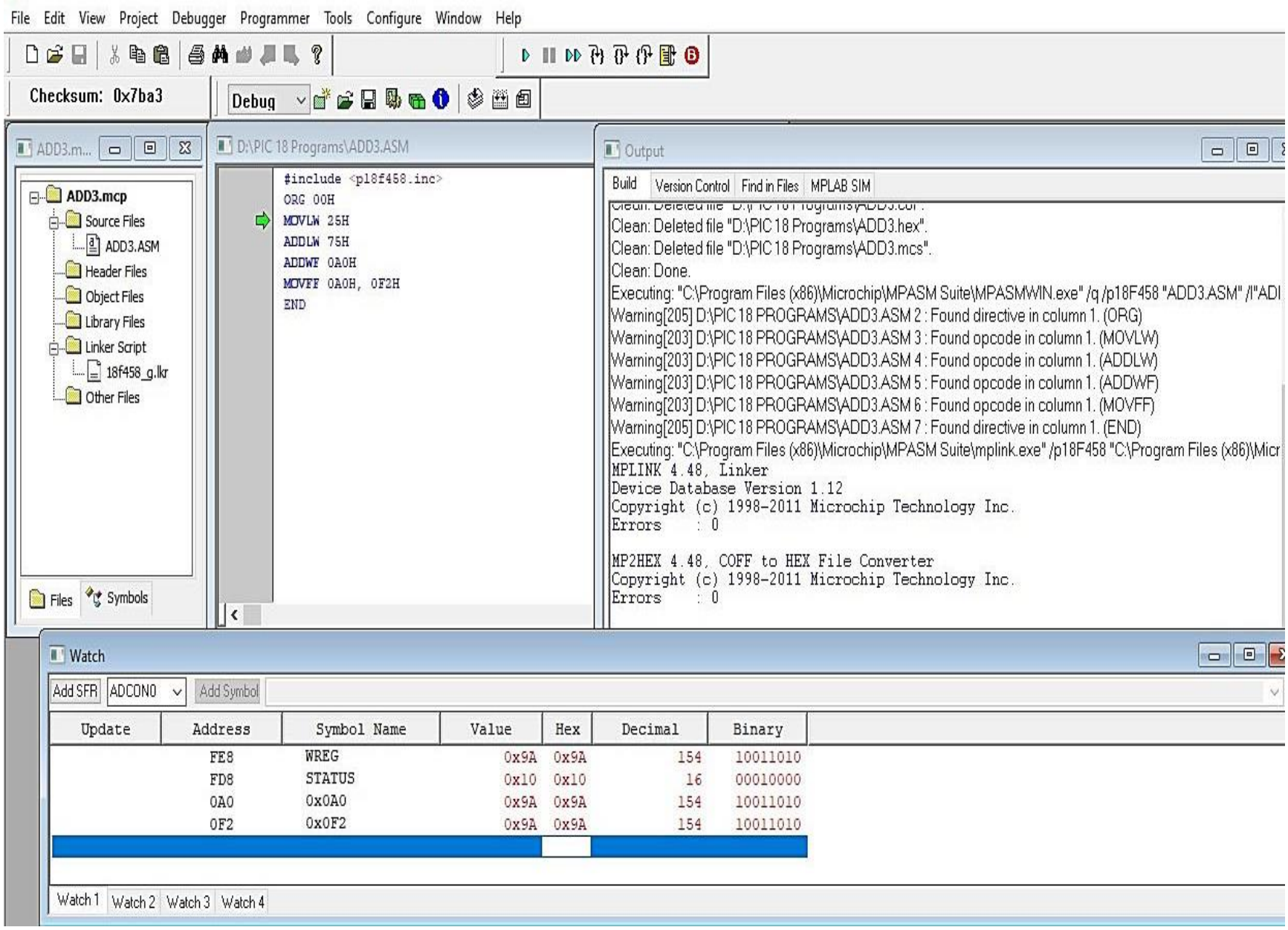
Algorithm:

1. Get first number in WREG.
2. Move value of W in any file loaction.
3. Add two numbers stored in WREG.
4. Check result of addition in STATUS.
5. Stop

Result: Value of WREG:

Value of Status Reg.:

Conclusion: Using assembly language programming 8 bit number is added and result is storedin WREG. Attached Output of program.


File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x7ba3
Debug
ADD3.m...
ADD3.mcp
Source Files
ADD3.ASM
Header Files
Object Files
Library Files
Linker Script
18f458_g.lkr
Other Files
Files
Symbols
D:\PIC 18 Programs\ADD3.ASM
#include <p18f458.inc>
ORG 00H
MOVLW 25H
ADDLW 75H
ADDWF 0A0H
MOVFF 0A0H, 0F2H
END
Output
Build
Version Control
Find in Files
MPLAB SIM
Clean: Deleted file "D:\PIC 18 Programs\ADD3.hex".
Clean: Deleted file "D:\PIC 18 Programs\ADD3.mcs".
Clean: Done.
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe" /q /p18F458 "ADD3.ASM" /l"ADI
Warning[205] D:\PIC 18 PROGRAMS\ADD3.ASM 2 : Found directive in column 1. (ORG)
Warning[203] D:\PIC 18 PROGRAMS\ADD3.ASM 3 : Found opcode in column 1. (MOVLW)
Warning[203] D:\PIC 18 PROGRAMS\ADD3.ASM 4 : Found opcode in column 1. (ADDLW)
Warning[203] D:\PIC 18 PROGRAMS\ADD3.ASM 5 : Found opcode in column 1. (ADDWF)
Warning[203] D:\PIC 18 PROGRAMS\ADD3.ASM 6 : Found opcode in column 1. (MOVFF)
Warning[205] D:\PIC 18 PROGRAMS\ADD3.ASM 7 : Found directive in column 1. (END)
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe" /p18F458 "C:\Program Files (x86)\Micr
MPLINK 4.48, Linker
Device Database Version 1.12
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0
MP2HEX 4.48, COFF to HEX File Converter
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0
Watch
Add SFR
ADCON0
Add Symbol
Update
Address
Symbol Name
Value
Hex
Decimal
Binary
FE8 WREG 0x9A 0x9A 154 10011010
FD8 STATUS 0x10 0x10 16 00010000
0A0 0x0A0 0x9A 0x9A 154 10011010
0F2 0x0F2 0x9A 0x9A 154 10011010
Watch 1
Watch 2
Watch 3
Watch 4

Experiment No. 2(B)

Aim: Write an assembly language program to subtract two 8 bit numbers.

Apparatus: MPLAB IDE software

Program statement:

Assuming two numbers are available in A & B resistor; write a programin assembly language of PIC18F458 to subtract two 8 bit numbers.

Explanation:

There are two 8 bit numbers. Clear Working register, load first number to Working register and second number to B register. Use SUBLW 87H instruction for subtraction. SUBLW subtracts the indicated variable and the carry flag together from the Working register, leaving the result in the Working register. SUBLW sets the carry (borrow) flag if a borrow is needed for bit 7 and clears C otherwise. (If C was set before executing a SUBLW instruction, this indicates that a borrow was needed for the previous step in a multiple- precision subtraction, so the carry is subtracted from the Working register along with the source operand.) AC is set if a borrow is needed for bit 3 and cleared otherwise. OV is set if a borrowis needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

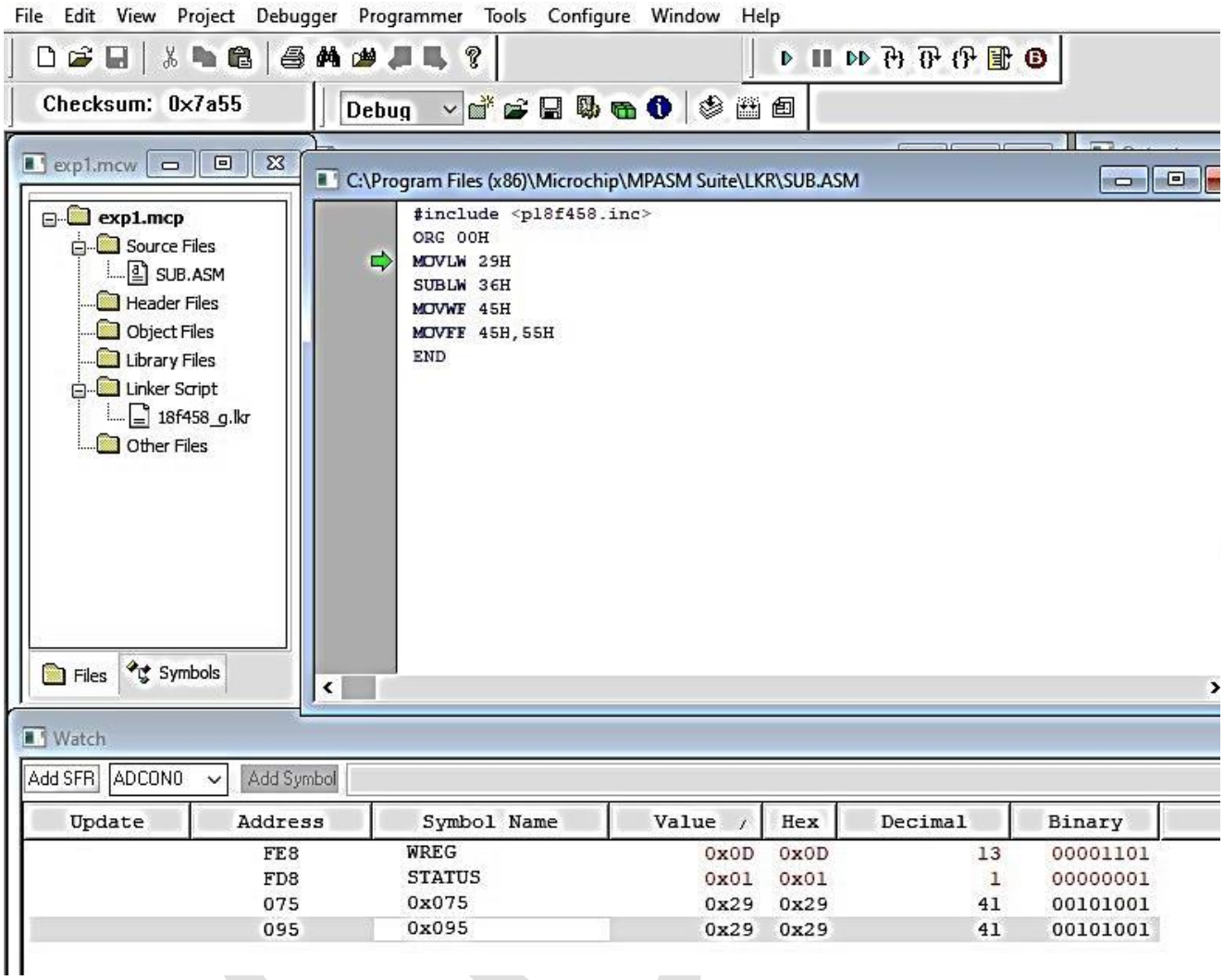
Algorithm:

1. Get first number in WREG.
2. Move value of W in any file loaction.
3. Subtract two numbers stored in WREG.
4. Check result of addition in STATUS.
5. Stop

Result: Value of WREG:

Value of Status Reg.:

Conclusion: Using assembly language programming 8 bit numbers are subtracted and result is stored in working register.


File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x7a55
Debug
exp1.mcw
exp1.mcp
Source Files
SUB.ASM
Header Files
Object Files
Library Files
Linker Script
18f458_g.lkr
Other Files
Files
Symbols
C:\Program Files (x86)\Microchip\MPASM Suite\LKR\SUB.ASM
#include <p18f458.inc>
ORG 00H
MOVLW 29H
SUBLW 36H
MOVWF 45H
MOVFF 45H,55H
END
Watch
Add SFR
ADCON0
Add Symbol
Update
Address
Symbol Name
Value
Hex
Decimal
Binary
FE8 WREG 0x0D 0x0D 13 00001101
FD8 STATUS 0x01 0x01 1 00000001
075 0x075 0x29 0x29 41 00101001
095 0x095 0x29 0x29 41 00101001

Experiment No. 2(C)

Aim: To write an ALP for multiplication of two 8 bit numbers.

Apparatus: - MPLAB IDE software

Program method:

Assuming that two number 8 bit in size.

Write a program of assembly language of PIC18F458to multiply two numbers.

Explanation:

There are two 8 bit numbers. Clear Working register, load first number to Working register and second number to file register. Use MULW 68H instruction for multiplication. MULW multiplies the unsigned 8-bit integers in the Working register and FILE register. The low-order byte of the 16-bit product is left in the PRODL, and the high-order byte in PRODH. If the product is greater than FFFFH, the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Algorithm:

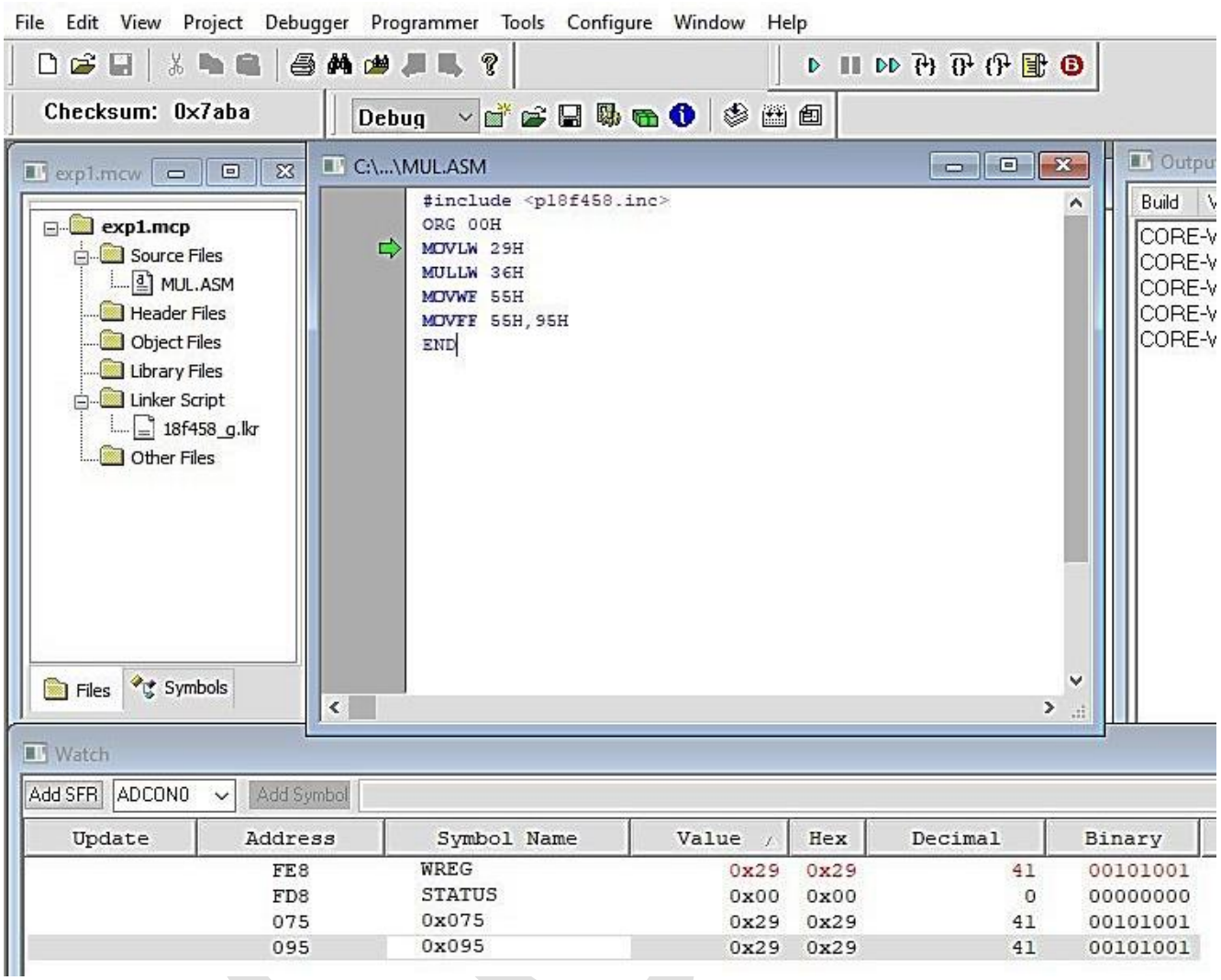
1. Get first number in WREG.
2. Move value of W in any file location.
3. Multiplication two numbers stored in PRODH and PRODL.
4. Check result of addition in STATUS.
5. Stop

Result: Value of WREG:

Value of PRODH Reg.:

Value of PRODL Reg.:

Conclusion: Using assembly language programming 8 bit number are multiplied and 16-bit product is left in the PRODL, and the high-order byte in PRODH.


File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x7aba
Debug
exp1.mcw
exp1.mcp
Source Files
MUL.ASM
Header Files
Object Files
Library Files
Linker Script
18f458_g.lkr
Other Files
Files
Symbols
C:\...\MUL.ASM
#include <p18f458.inc>
ORG 00H
MOVLW 29H
MULLW 36H
MOVWF 55H
MOVFF 55H,95H
END
Build
CORE-v
CORE-v
CORE-v
CORE-v
CORE-v
Watch
Add SFR
ADCON0
Add Symbol
Update
Address
Symbol Name
Value
Hex
Decimal
Binary
FE8 WREG 0x29 0x29 41 00101001
FD8 STATUS 0x00 0x00 0 00000000
075 0x075 0x29 0x29 41 00101001
095 0x095 0x29 0x29 41 00101001

Experiment No. 3

Aim : To study timer and delay programming for square wave.

Apparatus:

Sr. No.	Apparatus	Type	Qty
1	MPLAB IDE software	Simulator software	1

Learning Objectives:

Know timer/counter Know how to find value to be loaded into the timer.

Theory:

Timer 0: The timer0 module has the following feature timer / counter.

1) Software selectable as an 8 bit or 16 bit timer/counter
2) Readable and writable.
3) Dedicated 8 bit software programmable prescale.
4) Lock source selectable to be external to internal.
5) Interrupt on overflow from FFIT to 00IT
6) Edge select for external lock.

Register shows the timer 0 control register (T0CoN) figure shows a simplified block diagram of the timer module in 8 bit mode and figure 11-2 shows simplified block diagram of the timer 0 module in 16 bit mode the ICON register is a readable and write register that controls all the aspects timer 8 including the per of selection scales

BIT7	TOB	ToOC5	TOSE	PSA	TOPS2	TOPS1	BIT
TM Room	Bit						TOPSO

Bit 7: TMROON timer 0 on/off control

bit - 1=enable timer 0. 0 = Stops timer 0.

Bit 6: T0 8 bit timer 0, 8 bit / 16 bit control bit 1= timer 9 is configured as on 0 bit timer /counter 0 = timer 0 is configured as a 16 bit 1 timer / counter.

Bit 5: 70 C5: timer 0 clock source selected 1-transition on 700k 7 pin 0 = internal Instruction Cycle clock Bit 4: TOSE tiger o Source edge select be 1= increment on high to low transition on Tock I pin 0 = increment on low transition on Ly Tock I pin

BIT 4: PSA: timer o pre scalar assignment timer0 pre scalar is not assigned timer clocks input bypass pre scalar.) timer 0 prescalar is assigned timer 0input comes from pre scaler output.

Bit 2-0: T0PSO timer o pre scaler select bit.

111 = 1/250 pre scale value

110 = 1/128 pre scale value

101 = 1/64 pre scale value

100 = 1/32 pre scale value

011 = 1/16 pre scale value

010 = 1/8 pre scale value

011 = 1/2 pre scale value

Finding Scale value to be loaded into the timer XTAL=10MI = & no. finding the can use the following steps

Step 1. Divide the desired time delay by 0.4 micro sec to get count (n)

Step 2. Perform 65536-n, Where n is decimal value we got in step 1.

Step 3. Convert the result of step 2 to it ex YYXX .THIROH & TMROL register values.

1. Divide the desired time Step 1 microsec to get count (n) delay by 04
2. Perform 65536-n, where n is decimal value we got in step 1.
3. Convert the result of step 2 to Here YYXX which is the initial hex value to be loaded into the timer's register. Set TMROL = XX and TMROH=YY

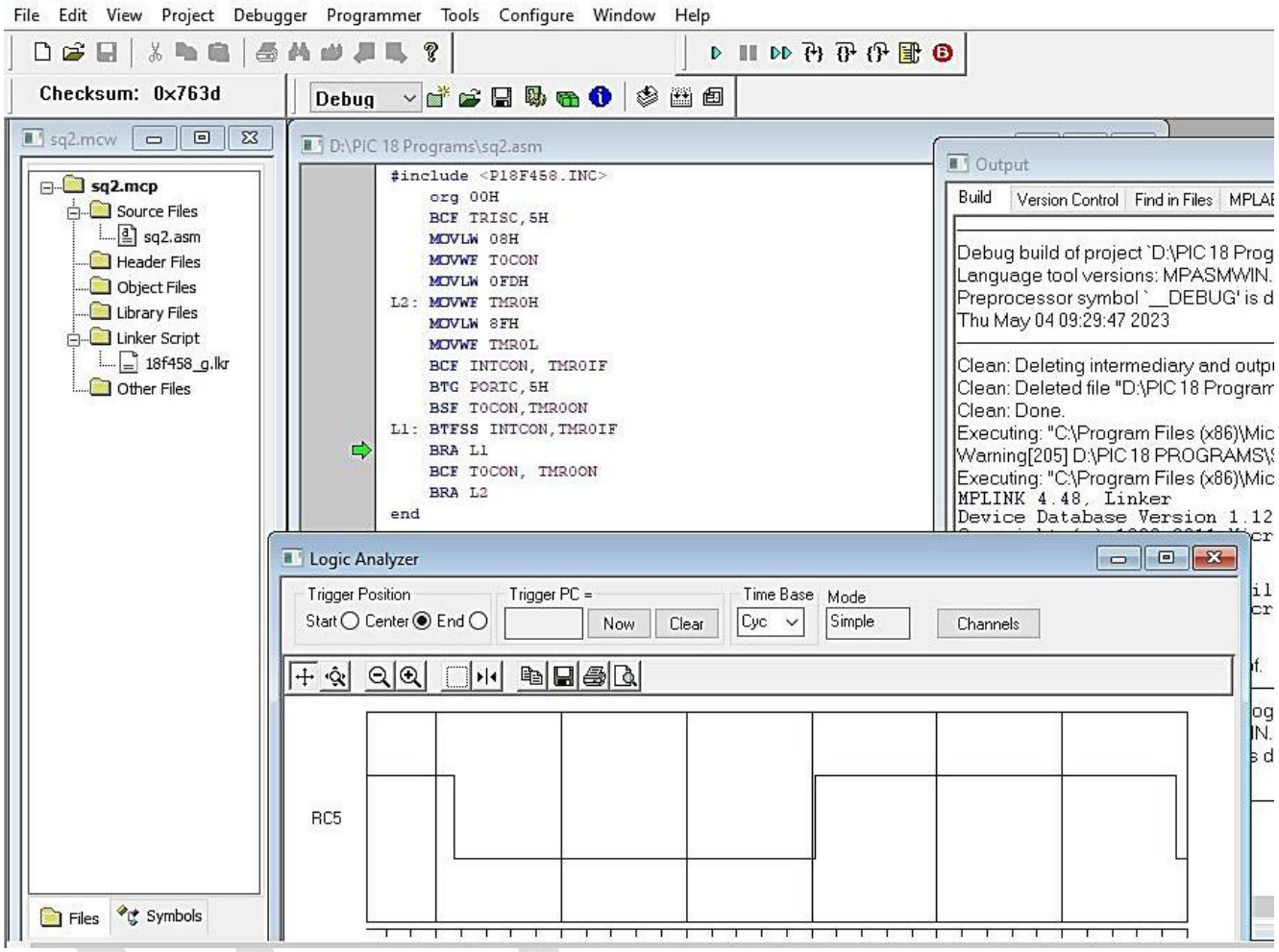
Program:

```
#include <P18F458.INC>
        ORG 00H
        BCF TRISC,5H
        MOVLW 08H
        MOVWF T0CON
        MOVLW 0FDH
L2: MOVWF TMR0H
        MOVLW 8FH
        MOVWF TMR0L
        BCF INTCON, TMR0IF
        BTG PORTC,5H
        BSF T0CON,TMR0ON
L1: BTFSS INTCON,TMR0IF
        BRA L1
        BCF T0CON, TMR0ON
```

BRA L2

END

Conclusion: Thus, We have studied timer 0 to create delay experiment.



Experiment No. 4

Aim: Transfer given data set to parts B

Apparatus:

Sr. No.	Apparatus	Type	Qty
1	MPLAB IDE software	Simulator software	1

Learning Objective

1) Know about different types of parts.
2) Know dual function of parts.
3) Know programming for data transfer to ports.

Theory: Depending on the device selected there are up to eleven ports available some pins to of the I/0 ports are multiplexed with an alternate function from the peripheral is enabled features on device in generate when a peripheral is enabled that pin may not be used as a general purpose I/0 pin each port has three register for its operation. These register are:

TRIS register (Data direction register) port register (Reads the level on the pin of LAT register the device (output latch). The data latch (LAT Register) is useful for read modify write operation on the value that the I/0 are driving

Initializing port B: Port B is an 8 bit wide bit directional port the corresponding data directionregister is TRISB setting a TRISB bit (=1) will make the corresponding output driver in a impedance mode) clearing a TRISB bit (=0) will make the corresponding port B an output i.e. put the content of the output latch on the selected pin read and modify write operation on the LATB register Read and write latched output value for port B.

PORT B Function:

Name	Bit	Buffer	Function
RB01 INT0	Bit 0	TTL1ST1	i /o pin 0 external ex interrupt 0 i/p internal software programmable weak put up.
RB01 INT	Bit1	TTL1 ST1	i/p and o/p or external interrupt and i/p internal software programmable weak put up.
RB21 CANTX	Bit2	TTL	i/p and o/p pin or can bus transmit pin.
RB31 CANTx	Bit3	TTL	i/p and o/p pin or CAN bus transmit pin.
RB4	Bit4	TTL	i/p and o/p pin or CAN bus Receive pin.
RB3 PGM	Bit5	TTL	i/p and o/p pin (with interrupt on charge internal software programmable weak put up low voltage serial programming enable.
RB31 PG6	Bit6	TTL	i/p and o/p pin (with interrupt on charge internal software programmable weak put up low voltage serial programming clock.
RB31 PGD	Bit7	TTL1 ST 2	i/p and o/p pin (with interrupt on charge internal software programmable weak put up low voltage serial programming data.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
LATB	LATB	Data	Output	Register				
TRISB	PORTB	Data	Direction	Register				
INTCON1	GIE/GIEN	PEIE/GIEL	TMROIE	INTOIE	TMR01F	TMR0F	INT01F	RBIF
INTCON2	RBPU	INTEDGO	INTEDG1	-	TMR01P	TMR01P	-	RBIP
INTCON3	INT2IP	INT1IP	-	INT21E	-	-	INT21F	INT1F

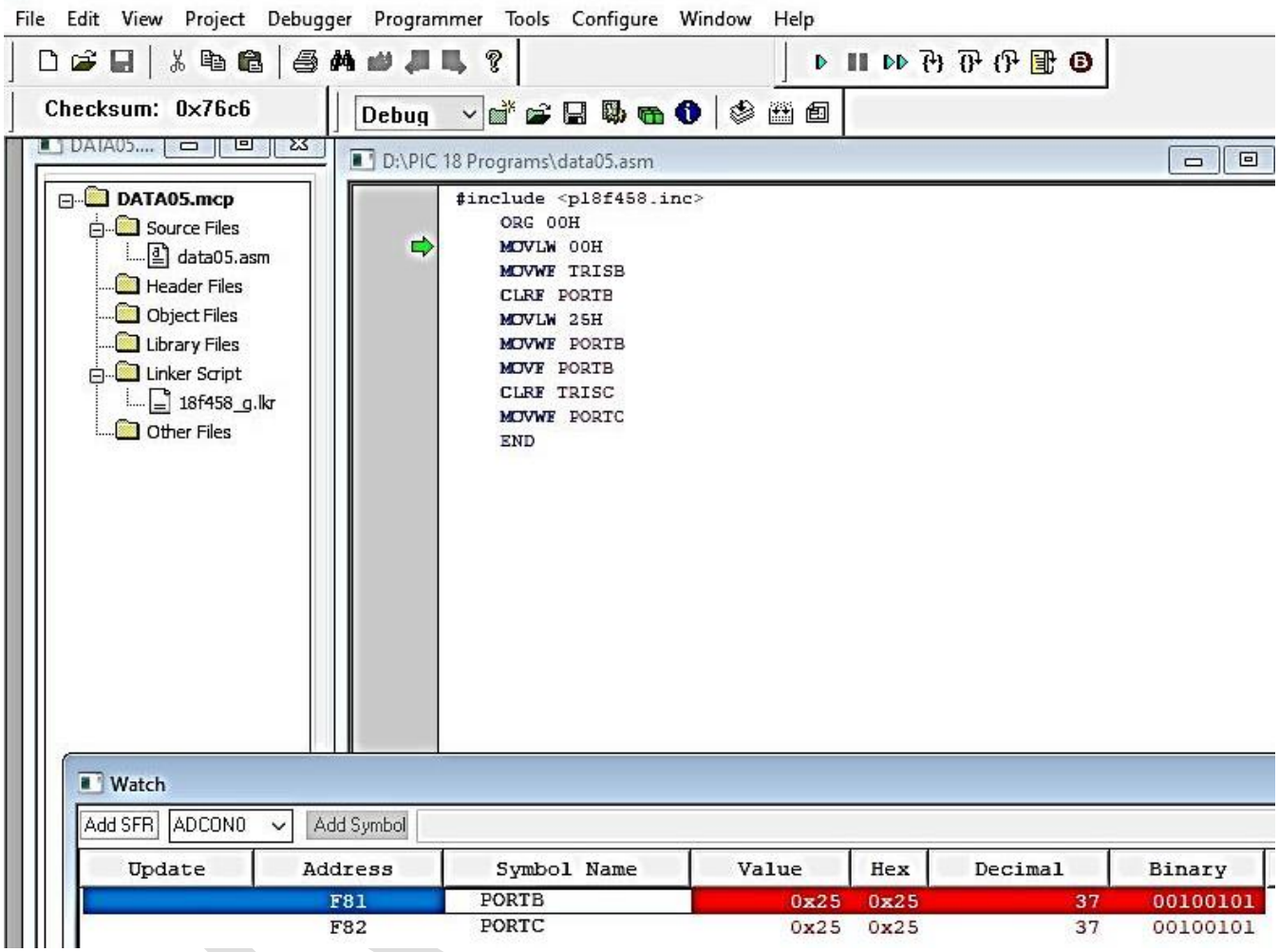
CLRF Port B : Initialize port B By Clearing output data latch.

CLRF LATB : Alternate method to Clear Output Data Latches.

MOVLW OXCF: Value used to initialize data direct.

MOVEF TRISB: Set RB3: RB0 as input: RB5:RB1: as output RB7: RB6 As input.

Conclusion: Thus we have studied transfer give data set to port B.



Experiment No. 5

Aim: Write program for microcontroller PIC 18F458 LED interfacing.

Learning Objective:

1. Give an overview of architecture and pin out of PIC18F.
2. Can create project using MPLAB IDE,C18 complier.
3. Change the delay between LED chasing.

Equipment's:

Sr. No.	Name	Type
1	Software MPLAB IDE,C18 complier	Software

Theory:

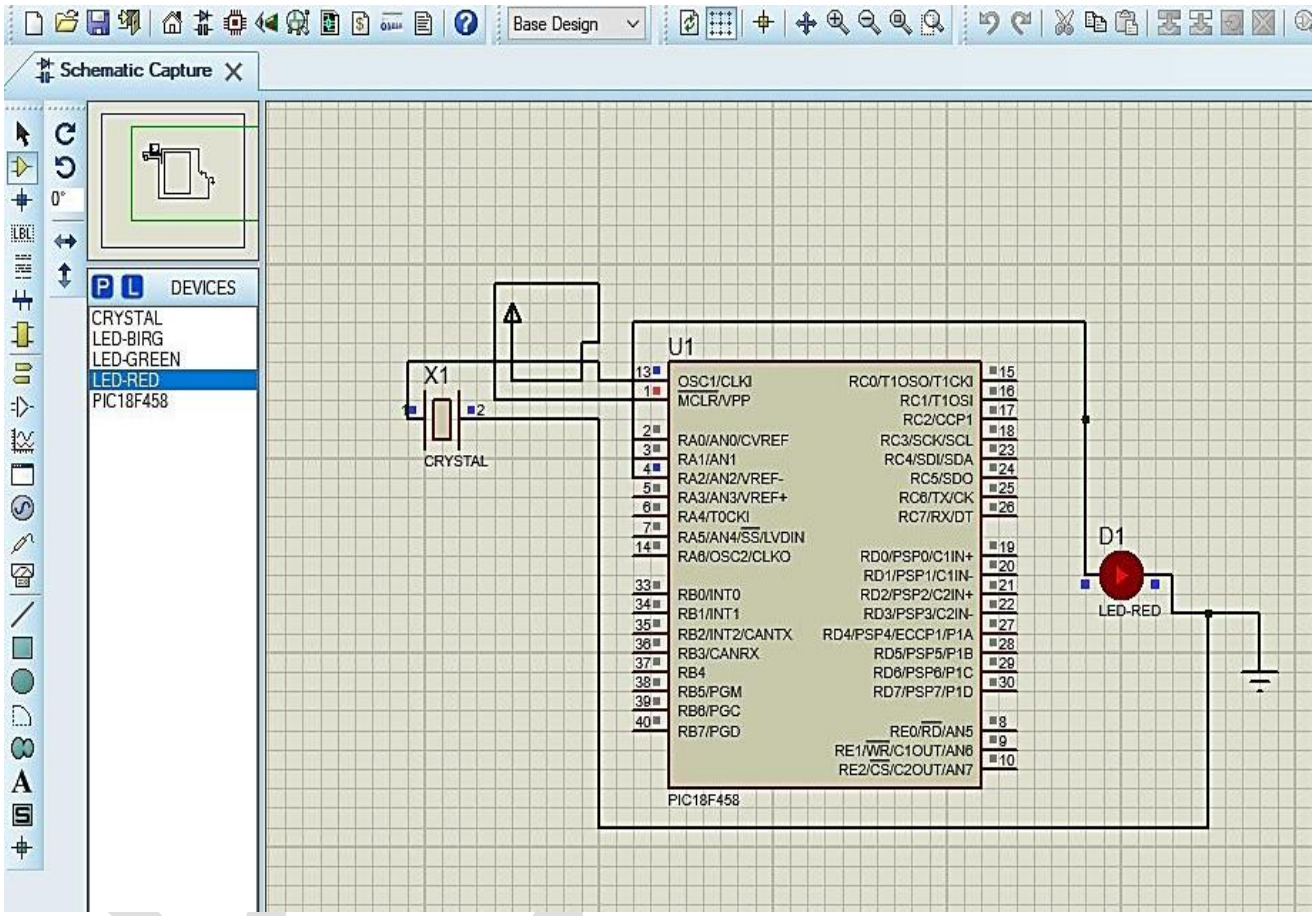
A light emitting diode (LED) is a two lead semiconductor light source. It is a basic PN-junction diode, which emits light when activated when a fitting voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of protons connections to the microcontroller are shown in diagram . The most familiar are of switch is a manually operated electromechanical device with one or more sets of electrical contact which are connected to external circuit. Each sets of contacts can be in one of two states either closed meaning the contacts are touching and electricity can flow between them or open means the contacts are separated and the switches may be operates by switch used to control a mechanism actuating the transition between these two set (open or closed) can be either a toggle (flip software for continuous ON or OFF) or momentary (Push for ON or PUSH for OFF) type.

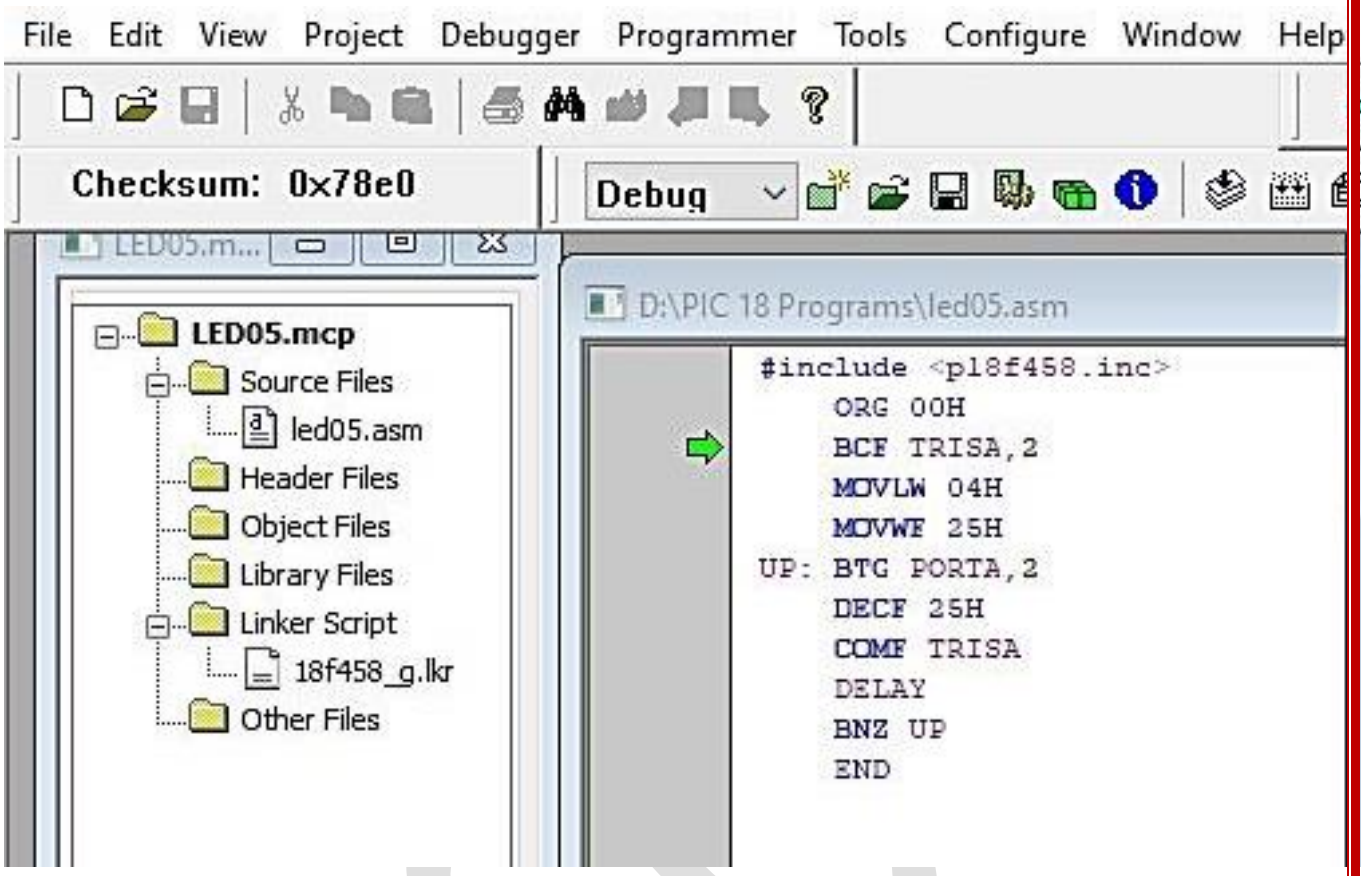
A switch may be directly manipulated by a human as a control signal to a system, such as a computer keyboard button, or to control power flow in a circuit such as light software automatically operated switches can be used to control the motion of machines for example to indicate that a garage door has reached its full open position or that a machine tool is in a position to accept another work piece ,switches may be operated by process variable such as pressure, temperature, flow current, voltage and force ,acting as sensors in a process and used to automatically control a system for example a thermostat is a temperature operated switch used to control a heating process. A switch that is operated by another electrical circuit is called relay.

Procedure:

1. Understand the problem statement and write it into algorithm steps.
2. Write C18 program in MPLAB and create hex file using C18 complier.
3. Download hex file using PIC loader into the PIC18F458 kit and observe the results.

Conclusion: Thus, we have studied LED interfacing experiment.




File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x78e0
Debug
LED05.mcp
Source Files
led05.asm
Header Files
Object Files
Library Files
Linker Script
18f458_g.lkr
Other Files
D:\PIC 18 Programs\led05.asm
#include <p18f458.inc>
ORG 00H
BCF TRISA,2
MOVLW 04H
MOVWF 25H
UP: BTG PORTA,2
DECF 25H
COMF TRISA
DELAY
BNZ UP
END

Experiment No. 6

Aim: Write a Program for microcontroller PIC18F458 for DC motor interfacing.

Learning Objectives:

1. Know basic operation of DC motor.
2. Know different types of DC motor.
3. Know how to interface DC motor with PIC18F458.

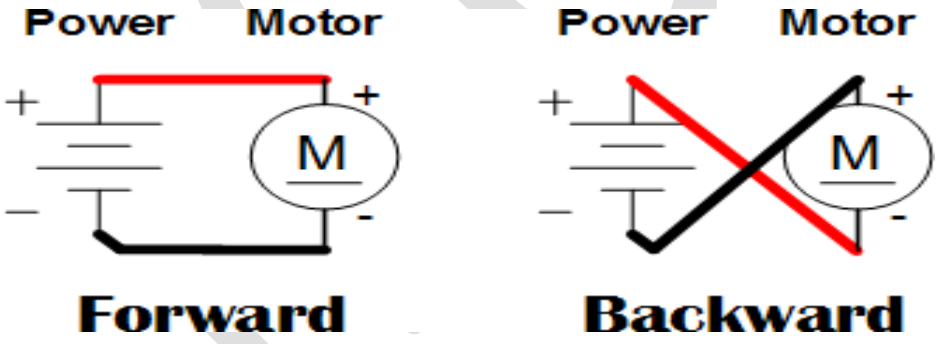
Equipment's:

1. Hard ware – PIC18F458 kit, DC motor , a driver circuit.
2. Software – MCC complier , MPLAB complier, simulator software.

Theory:

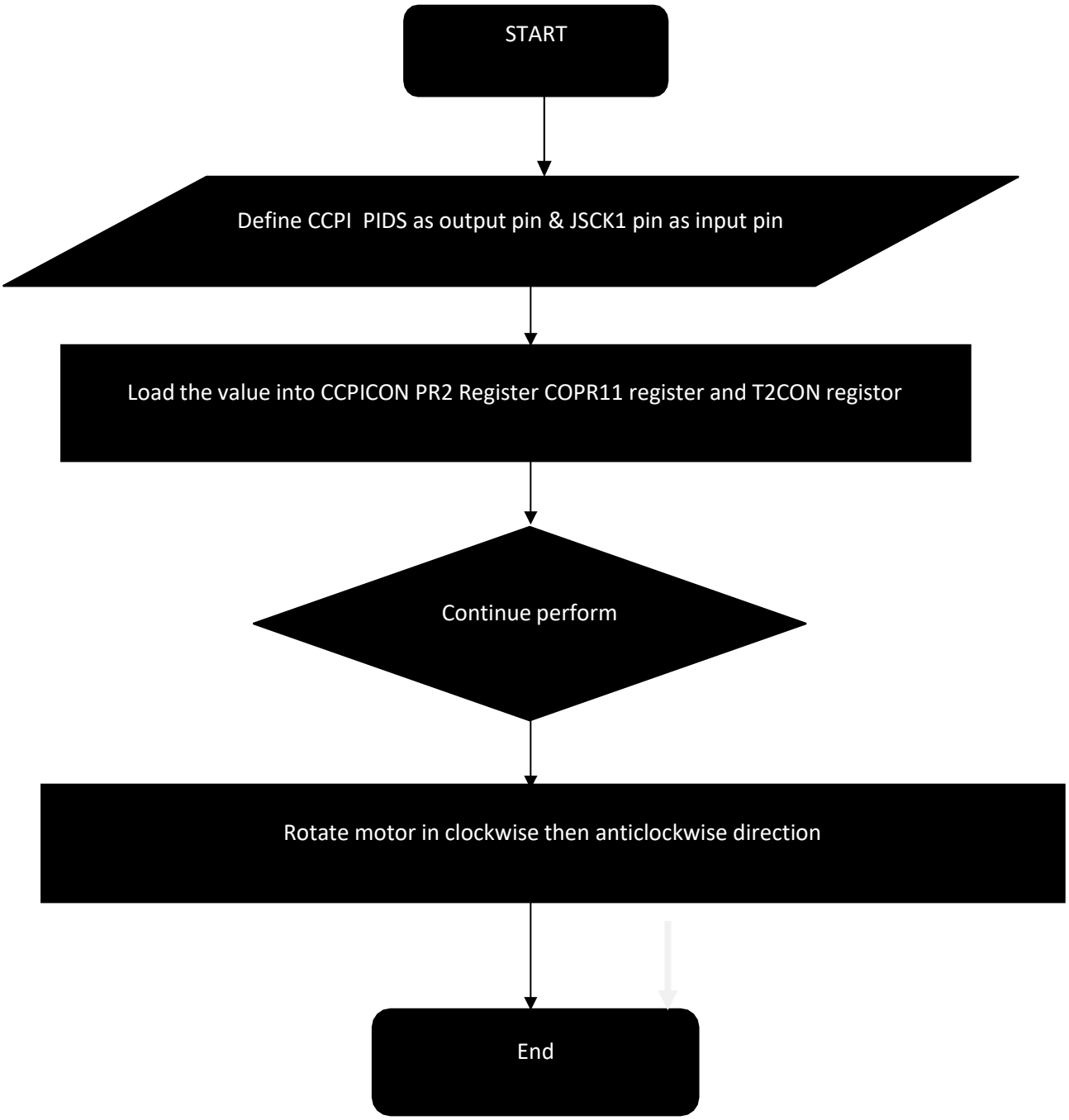
A Dc motor is another widely used device that translated electrical pulse into mechanical movement in the DC motor we have only + and – leads connecting them to a Dc voltage source moves the motor in one direction. By reversing the polarity. The DC motor will move in opposite direction. The speed of motor depends on 3 factors:

a) Load
b) Voltage
c) Current



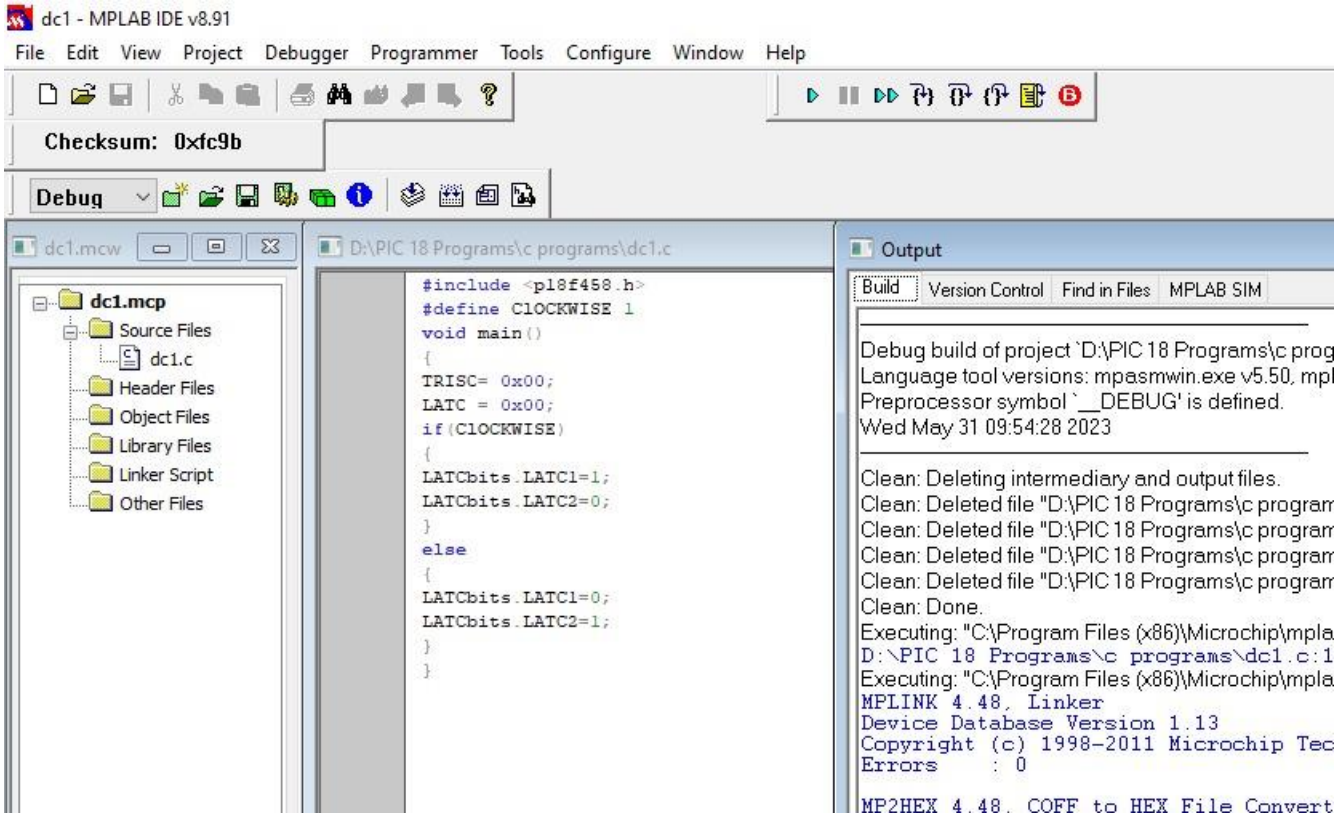
For a given fixed load it can maintain a steady speed by using a method called pulse width modulation (PWM) by changing (modulating) the width of the pulse applied to a Dc motor we can increases or decrease the amount of power provided to the motor thereby increasing or

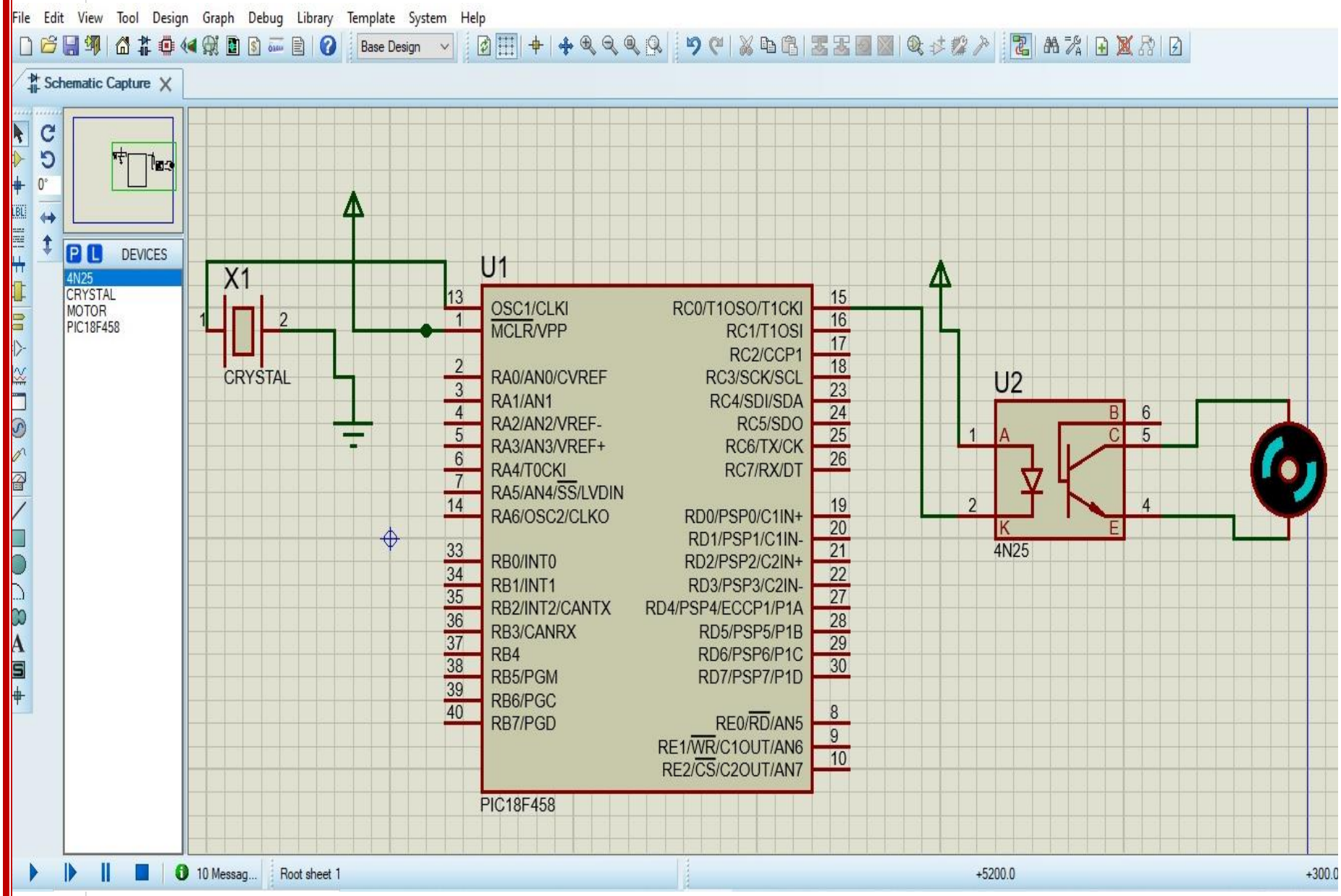
decreasing the motor speed although the voltage has fixed amplitude it has a variable duty cycle that means the wider the pulse the higher the speed PWM is so widely used in Dc motor control that some microcontroller comes with the PWM circulatory embedded in the strip.



Conclusion:

It gives study about interfacing of DC motor with PIC18F458 studies use of CCP module CCPand duty cycle registers.




File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
DEVICES
4N25
CRYSTAL
MOTOR
PIC18F458
X1
CRYSTAL
U1
OSC1/CLKI
MCLR/VPP
RA0/AN0/CVREF
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
RA6/OSC2/CLKO
RB0/INT0
RB1/INT1
RB2/INT2/CANTX
RB3/CANRX
RB4
RB5/PGM
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0/C1IN+
RD1/PSP1/C1IN-
RD2/PSP2/C2IN+
RD3/PSP3/C2IN-
RD4/PSP4/ECCP1/P1A
RD5/PSP5/P1B
RD6/PSP6/P1C
RD7/PSP7/P1D
RE0/RD/AN5
RE1/WR/C1OUT/AN6
RE2/CS/C2OUT/AN7
PIC18F458
U2
A
K
B
C
E
4N25
10 Messag...
Root sheet 1
+5200.0
+300.0

Experiment No. 7

Aim: Write a program for microcontroller PIC18F458 for stepper motor interfacing.

Learning objectives:

1. Know basic operation of stepper motor and its application.
2. Know different types of the stepper motor.
3. Know how to interface stepper motor with PIC18.

Equipment's:

1. Hardware – PIC18F458 kit stepper motor driver circuit.
2. Software – MCC complier, MPLAB complier, Simulator software.

Theory:

The stepper motor is advice which converts electrical pulse into mechanical continuously the stepper motor motor moves in steps of fixed size the most common steps motor has four stator winding that are paired with centre tapped common such type of stepper motor is known as four pulse or unipolar stepper motor. Stepper motor also known as stepping or step up motors and are essentially incremental motion. In applications such as disk drive dot matrix printers and robotics the stepper motor is used for position control.

Interfacing diagram:

PIC18F458	Value
RC0	05H
RC1	06H
RC2	21H
RC3	22H

Full step operation of stepper motor:

RC0	RC1	RC2	RC3
A	A’	B	B’
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

Stepper motor commonly has a permanent magnet rotor surrounded by the stator. Step angle is defined as the minimum degree of rotation is associated with of stepper motor is a single step the common step of the motor angle of the stepper motor is 1.8 degree 7.5 degree and 15 degree if rotates 90 degrees in each step it is called ‘full step’.

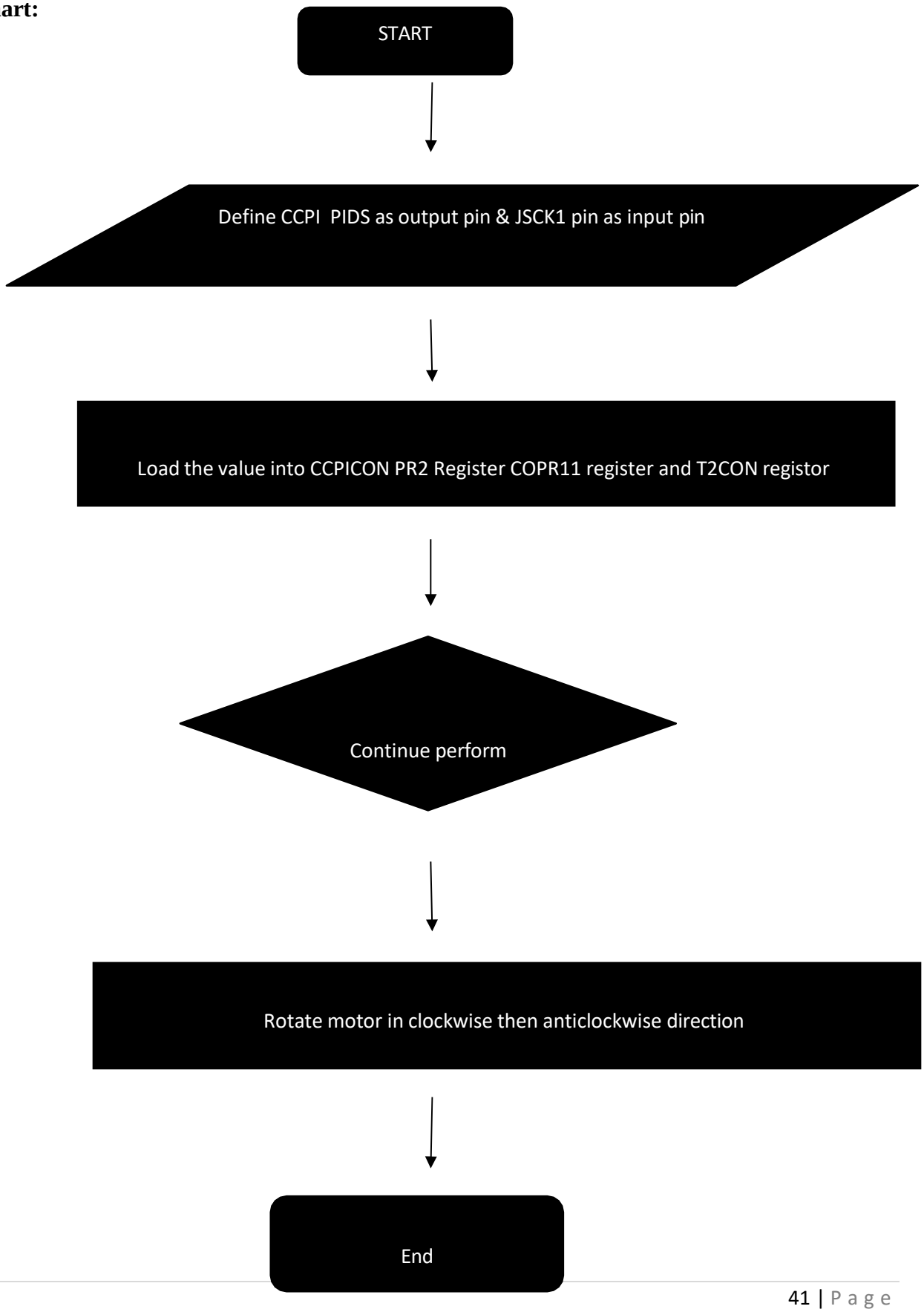
If step angle (step size) 1.8 degree then complete one rotation (360 degree) it require 200 step

Step angle	Step per revolution
0.72	500
1.8	200
2.0	180
2.5	144
5.0	72
7.5	48
15	24

Interfacing of stepper motor with microcontroller PIC18F458:

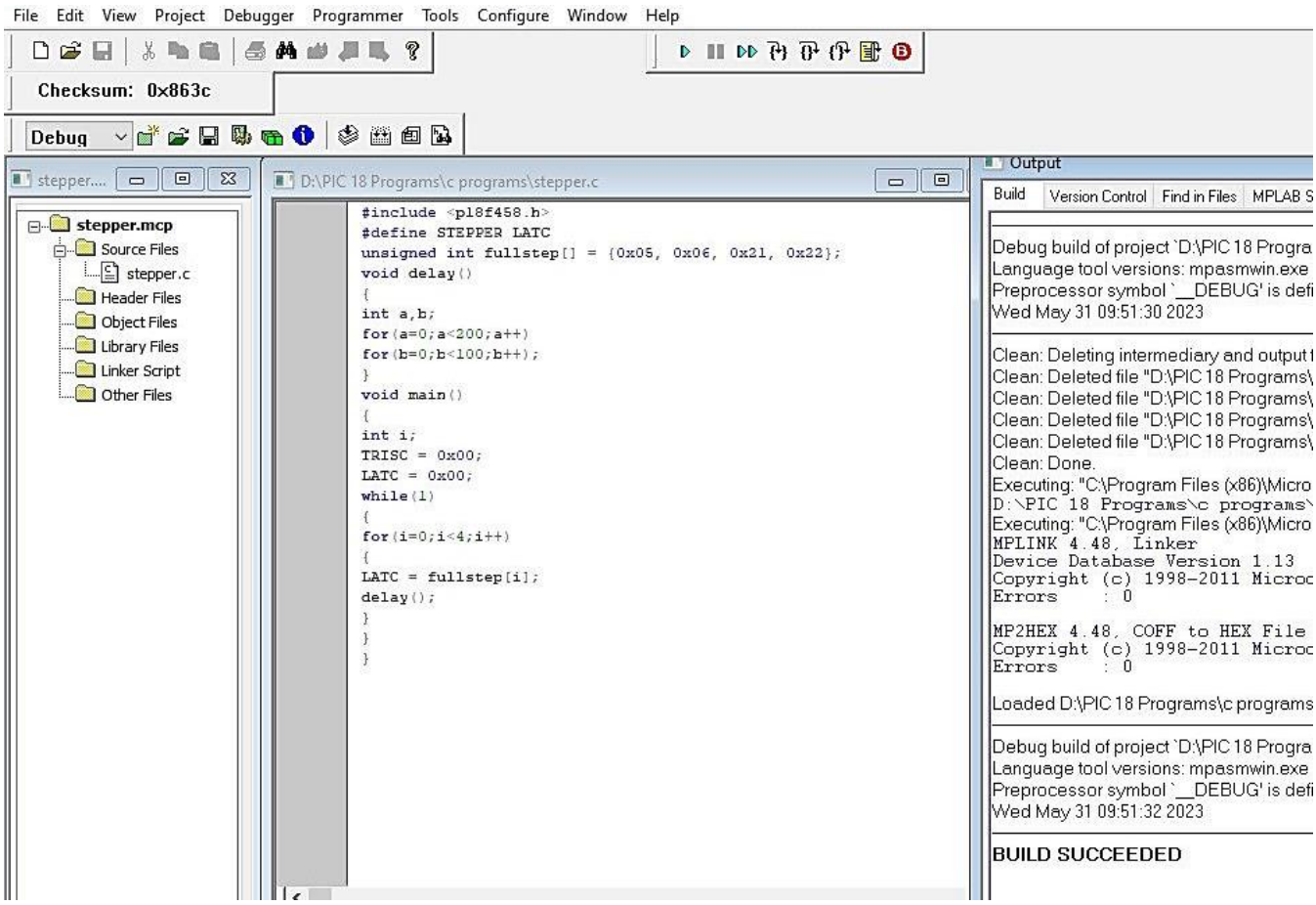
The stepper motor does not have sufficient current to drive stepper motor windings hence transistors are used to supply high current to the motor. The diodes are used to protect diodes from reverse biased. Here ULN2003 to energized stator winding.

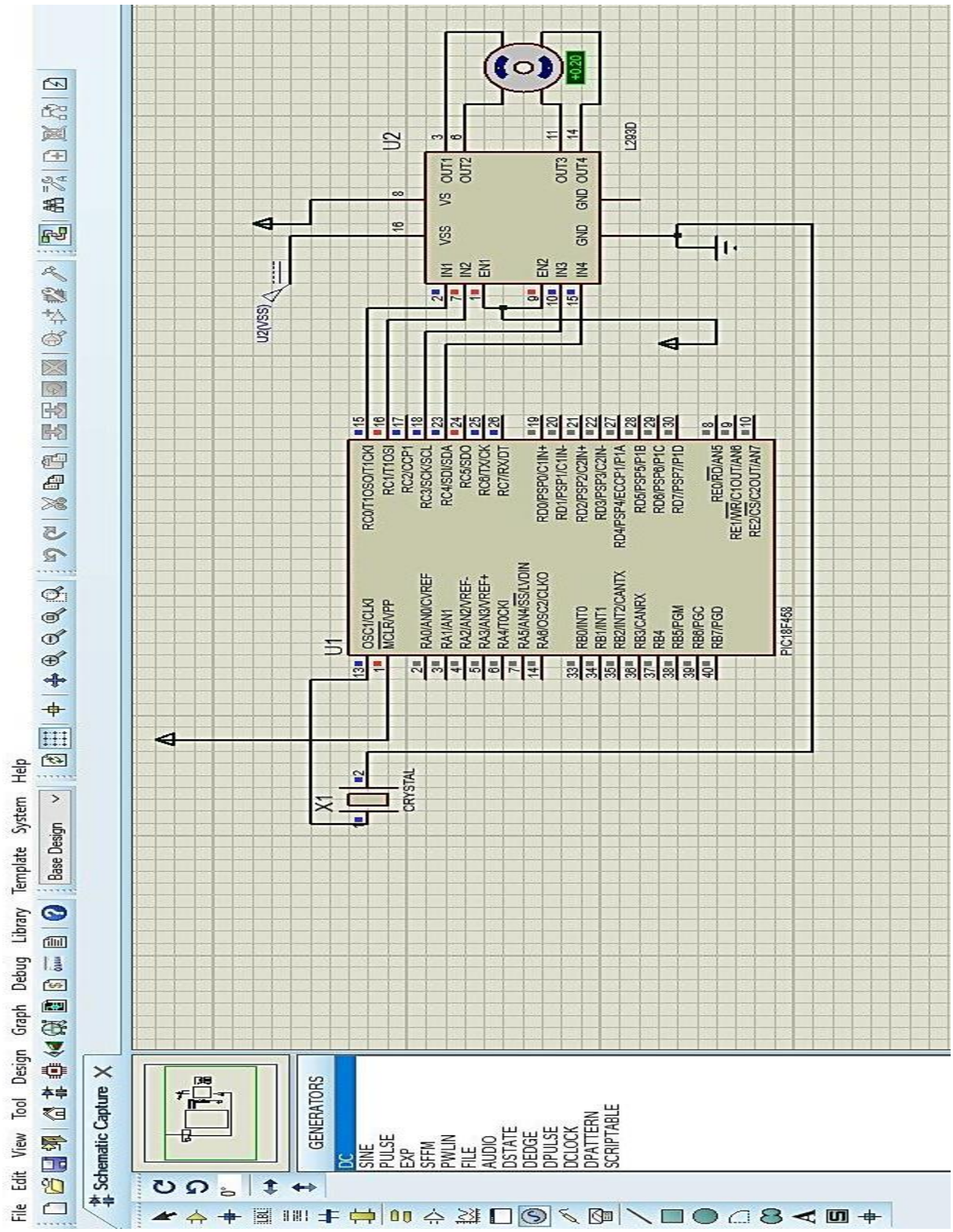
Flowchart:


START
Define CCPI PIDS as output pin & JSCK1 pin as input pin
Load the value into CCPICON PR2 Register COPR11 register and T2CON registor
Continue perform
Rotate motor in clockwise then anticlockwise direction
End

Conclusion:-

If studied interfacing of PIC18F458 with stepper motor driver circuit is required between microcontroller and stepper motor as microcontroller doesn't provide sufficient current to drive the stepper motor.




File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
GENERATORS
DC
SINE
PULSE
EXP
SFFM
PWLIN
FILE
AUDIO
DSTATE
DEDGE
DPULSE
DCLOCK
DPATTERN
SCRIPTABLE
U1
X1
CRYSTAL
PIC18F458
U2
L293D
U2(VSS)
+0.20

Experiment No. 8

Aim: To interface relay with PIC18F458 microcontroller.

Learning objectives:

1. know basic operation of LCD and its applications.
2. know how to interface relay with microcontroller PIC18F458

Equipment: -

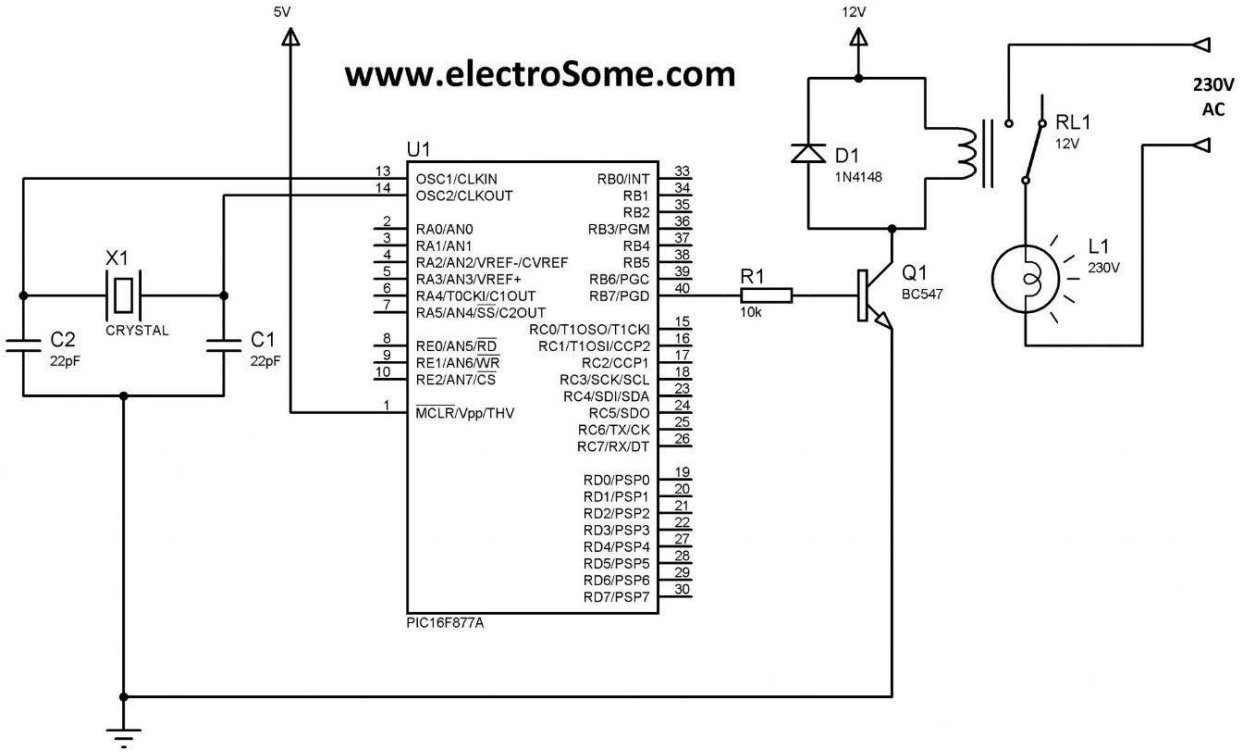
Software: MPIAB software and Porteous.

Theory: A relay is an electrically controllable switch widely used in industrial controls automobiles, and application. It allows the isolation of two separate sections of a system with then different voltage Sources for example, a +5V system can be isolated 120V system by placing a relay between them One such relay is called electromechanical (or electromagnetic) relay (EMR) as shown in Fig. The EMR have three components. The coil, spring and contact in fig a digital +5V on the left side can control a 12V motor on the right side with -out any physical contact between them. When current flows through the coil a magnetic field is created around the coil (the call is energized) which causes the armature to be attracted to the coil The armature contact act like a Switch and closes at opens the circuit when the call is not energized. a spring pulls the armature to be attracted to the coil. The armature contact acts a spring pulls the armature to its normal state of open or closed the block diagram for electromechanical relay we do not show the spring but it does exist internally. There is all type of relays for all kinds of applications. In choosing a relay the following characteristics need to be considered

1. The contacts can be normally open (NO) or normally closed (NC) In the NC type, the contacts are closed when the call is not energized in the NO, and the contacts are open when the coil is unenergized.
2. There can one or more contacts for example, we can have SPST (Single pole single throw). SPDT (Single pole, double throw) and DPDT (double pole double throw) relays
3. The voltage and current needed to energize the coil. The voltage can vary from a few volts to 50V while the current can be from a few mA to 20 mA. The relay has a minimumvoltage. below which the coil will not be energized This minimum vtg is called the “pull in” voltage in the datasheet for relay we might not see current but rather coil resistance the V/R will give you the pull in current. For example if the call voltage is 5V, and the coil resistance is 500 ohm, we need a minimum of 10 mA (5V/500 ohm = 10mA) pull in current.

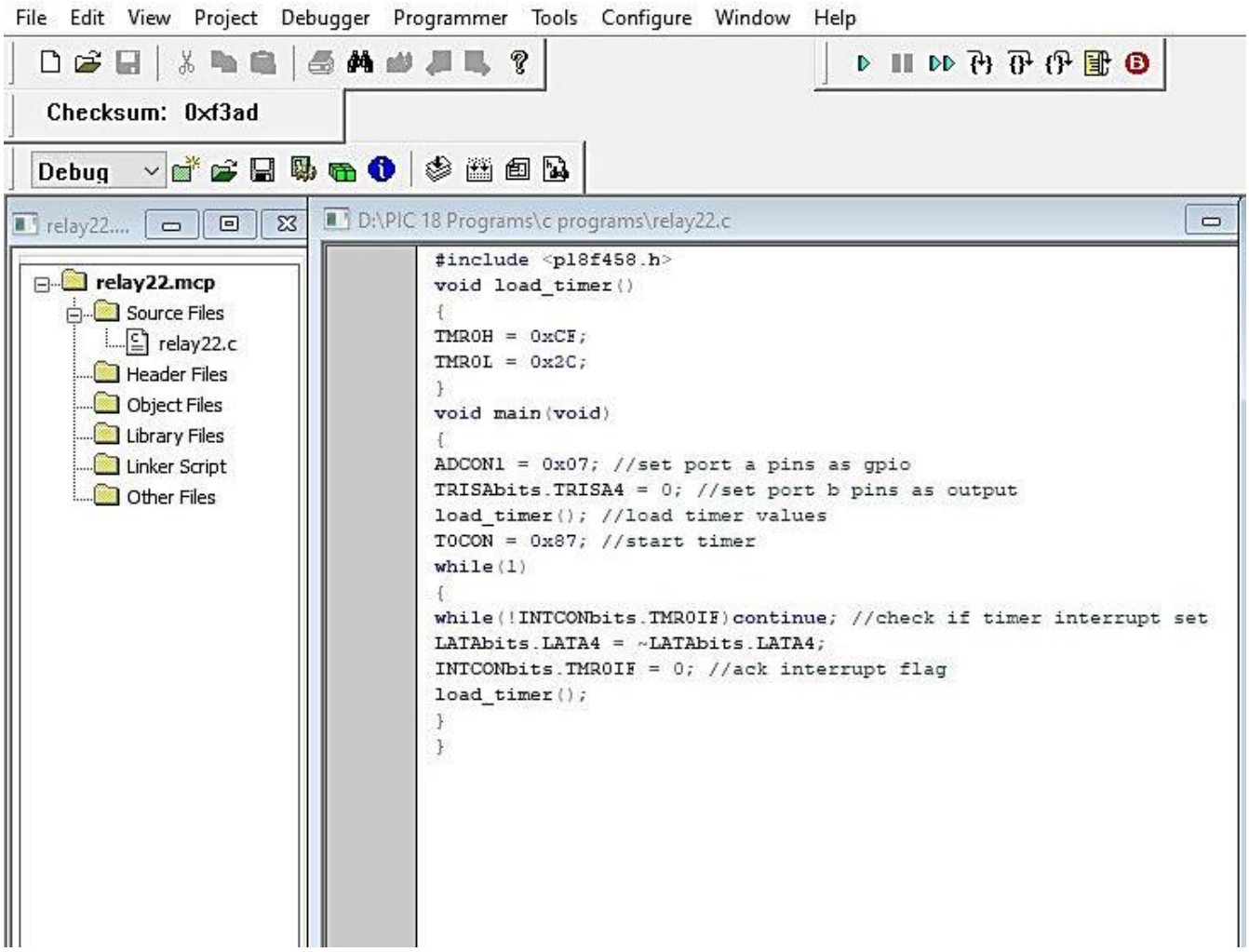
4. The maximum DC/AC voltage and current that can be handled by the contacts. This is in the range of a few volt to hundreds of volts, while the current can be from. Few amp to 40A or more, depending on the relay Notice the difference between this. voltage current specification and the vtg / current needed for energizing the coil. The fact that one use sucha small amount of vtg / current on one side to handle a large amount of vtg / current on the other side is what makes relay so widely used in industrial controls.

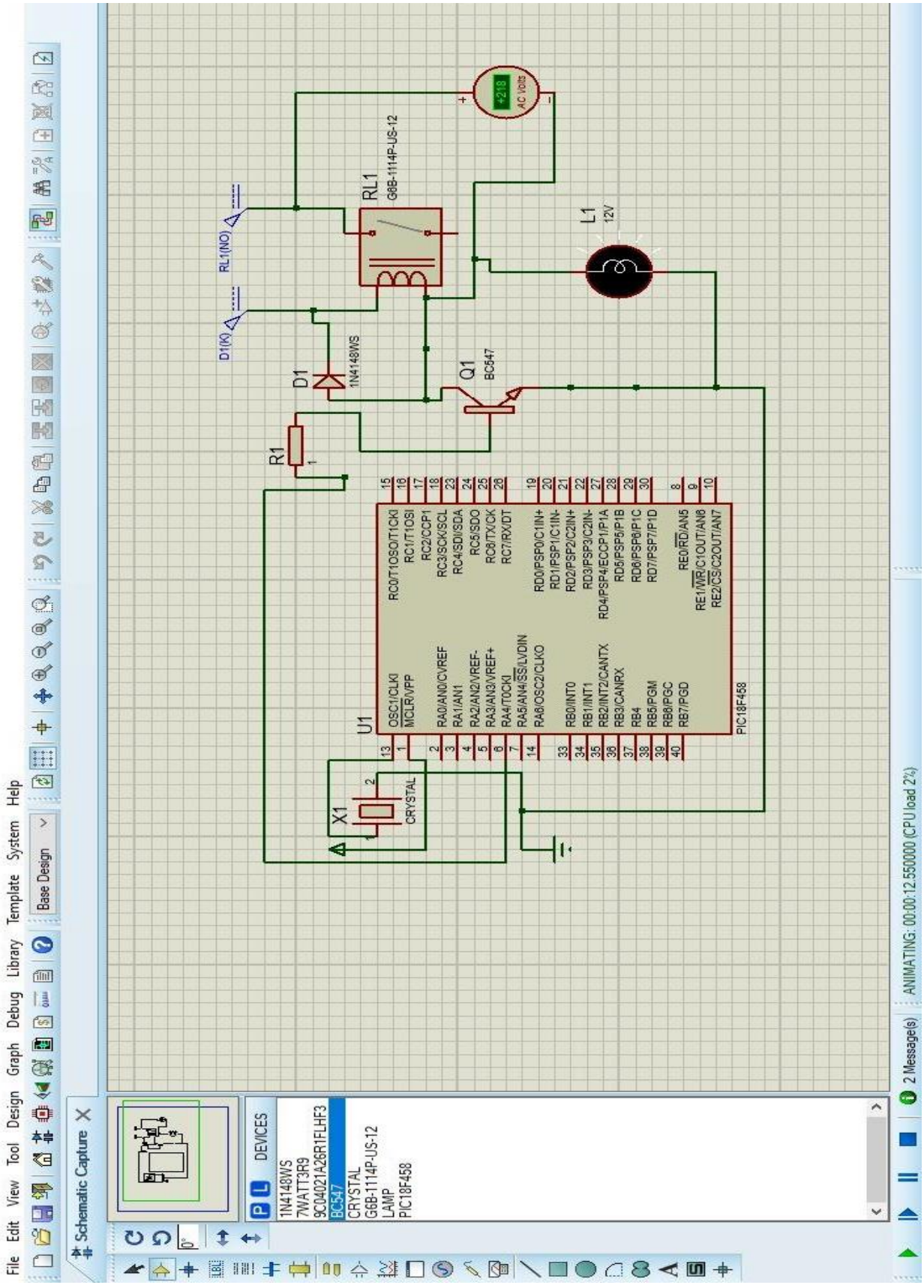
The relay is provided on the microcontroller PIC18F458 board for user to externally interface a high voltage circuit and also to provide isolation. The relay is connected to the PIC of pin RA4.



Driving a relay: Digital system and microcontroller pins lock Sufficient current to drive the relay while the relay coil needs around 10mA to be energized. the microcontroller pins can provide maximum. of 1-2 mA current for this reason a driver such as transistor is placedbetween microcontroller and the relay.

Conclusion: In this experiment we have successfully interfaced relay with PIC18F458 microcontroller.


File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0xf3ad
Debug
relay22....
D:\PIC 18 Programs\c programs\relay22.c
relay22.mcp
Source Files
relay22.c
Header Files
Object Files
Library Files
Linker Script
Other Files
#include <p18f458.h>
void load_timer()
{
TMR0H = 0xCF;
TMR0L = 0x2C;
}
void main(void)
{
ADCON1 = 0x07; //set port a pins as gpio
TRISAbits.TRISA4 = 0; //set port b pins as output
load_timer(); //load timer values
T0CON = 0x87; //start timer
while(1)
{
while(!INTCONbits.TMR0IF)continue; //check if timer interrupt set
LATAbits.LATA4 = ~LATAbits.LATA4;
INTCONbits.TMR0IF = 0; //ack interrupt flag
load_timer();
}
}


File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Base Design
DEVICES
1N4148WS
7WATT3R9
9C04021A26R1FLHF3
BC547
CRYSTAL
G6B-1114P-US-12
LAMP
PIC18F458
U1
X1
CRYSTAL
R1
D1
1N4148WS
Q1
BC547
RL1
G6B-1114P-US-12
RL1(NO)
D1(K)
L1
12V
+213
AC Volts
PIC18F458
2 Message(s)
ANIMATING: 00:00:12.550000 (CPU load 2%)

Experiment No: 9

Aim: Write a program for microcontroller PIC18F458 for LCD interfacing.

Learning Objectives:

1. Know basic operations of LCD and its application.
2. Know different LCD commands.
3. Know how to interface LCD with PIC18F458.

Theory:

In recent years LCD has been finding wide spread use replacing LED's seven segment LED or other multi segment LED's this is dual to following reasons.

1. The declining price of LCD
2. The ability to display numbers, characters and graphics this is in contrast to led's which are limited to number and few characters.
3. In corporation of a refreshing the LCD in contrast the LED must be refreshed by the CPU (or in some other way) to keep displaying the data.
4. Ease of programming for character and graphics.

LCD pin descriptions: The LCD discussed in this section has 14 pins the function of each pin is given in table fig shows the pin position for various LCD. VCC , VSS and VEE While Vcc and Vss provide +5v and ground respectively VEE is used for controlling LCD contrast.R/W, Read/ Write R/W input allows user to write information to LCD or read information from R/W=1 when reading and R/W = 0 when writing. Rs. Register select there are two very important register inside the LCD the RS pin is used for this selections as follows of Rs=0 the instruction command code register is selected allowing to use the same data display on LCD.

E, Enable: The enable pin is used for the LCD for latch in the information to LCD or read the contact of LCD internal when data is supplied to En pin in order for the LCD to latch in the data present at the data pins. This pulse must be minimum of 450 ns wide. In the look we call this delay SDELAY (short delay) to distinguish it from other delay.D0 – D7 = The 8 bit data pin D0-D7 are used to send information to LCd or read the contact of LCD internal register to display letters and numbers are send ASCII codes for the letter A-Z, a-2 and number 0-9 to their pins while Rs=1 There are also instruction comment code that can be send to LCD to clear the display or force the arrow to the home position as link to the cursor table below lists instruction command code t send any of the command instead in table to the LCD make pin Rs=0 for that data make pin Rs=1 the same for high

to low pulse to C pin to enable internal latch of LCD there are two ways to send characters (comments data to the LCD).

1. Use a delay before sending the next one.
2. Use the carry flag to see if the LCD is ready for the next one.

Symbol	I/O	Description
VSS	-	GND
VCC	-	Pow+5V power supply control
VEE	-	Power supply to control contrast.
RS	I	RS=0 comment register Rs=1 data register.
R/W	I	R/W=0 write R/W=1 for read end
E	I/O	8 bit data bus
DB0	I/O	8 bit data bus
DB1	I/O	8 bit data bus
DB2	I/O	8 bit data bus
DB3	I/O	8 bit data bus
DB4	I/O	8 bit data bus
DB5	I/O	8 bit data bus
DB6	I/O	8 bit data bus
DB7	I/O	8 bit data bus

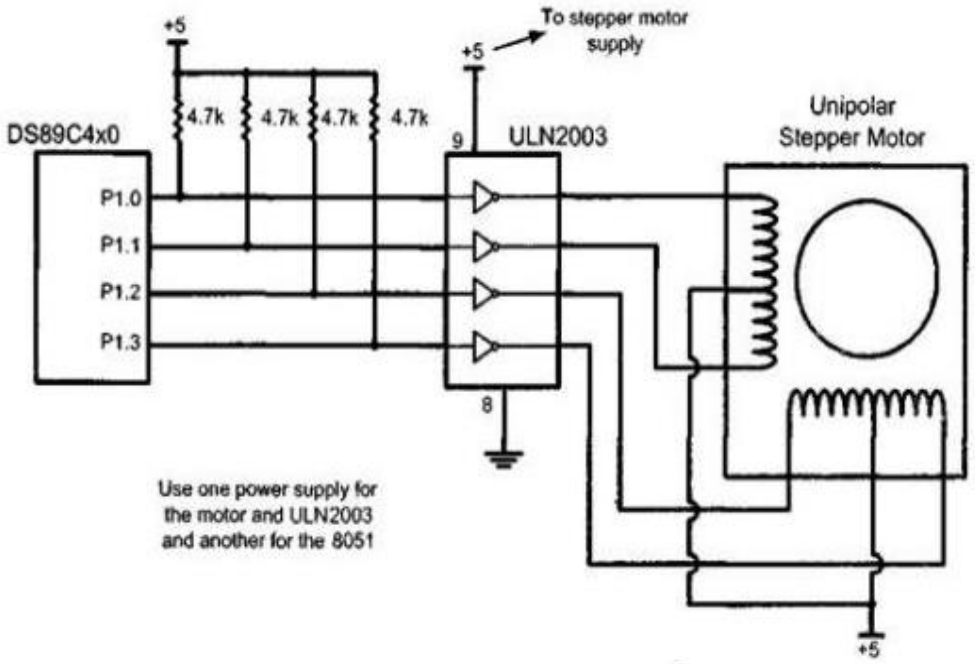


Fig: Interfacing diagram of stepper motor

Conclusion: Interfacing of LCD with PIC18F458. I studied with display of character on it.

Experiment No: 10

Aim: Write a C program for Lm35 interfacing with microcontroller PIC18F458.

Learning Objectives:

1. Know basic operation of Lm 35.
2. Know different types of the stepper motor.
3. Know how to interface Lm35 with PIC18F458

Equipment:

1. **Hardware**: PIC18F458 kit Lm35
2. **Software:** C18 complier/MPLAB complier/simulator, software.

Theory: Signal conditioning is widely used in the world of data the most common transducers produce an output in the form of voltage, current, charge, capacitance and resistance we need the to convert these signals into voltage however in order to send input to an A to D converter this conversion is commonly used signal conditioning can be converted a current to voltage conversion or signal amplification for e.g the thermistor changes resistance with temperature the change of resistance must be translated into voltage in order to be of any use to an ADC look at the case of connecting an LM35 to on ADC of the PIC18F458.

ADCON O Register :- The ADCON register is used to set the conversing timer and select the analog input channel among other things. In order to reduce the power consumption of PIC18 the ADC feature is turned off When we turn on the ADC bit of ADC register the other important bit is the G01 Done bit we use this bit to start conversion has ended. Notice in ADCONO that not all family members have all 8 analog input channel the conversion timer is set with the ADC's bits while ADCS bits. While ADCS and ADCSO holds the ADCONO register ADCS2 is in the post of ADCON1 register.

ADCON Register: Another major register of the PIC18 ADC feature is ADCON1 the ADCON1 register is used to select Break voltage among other things. It is shown in fig after the A/D conversion is complete the result sits in register ADRESL.(A?D result lower byte) and AAND ADRESH (A/D result higher byte) adfm bit of ADCON 1 is used for making it right justified or left justice because we need only 10 bits of the 16 bit.

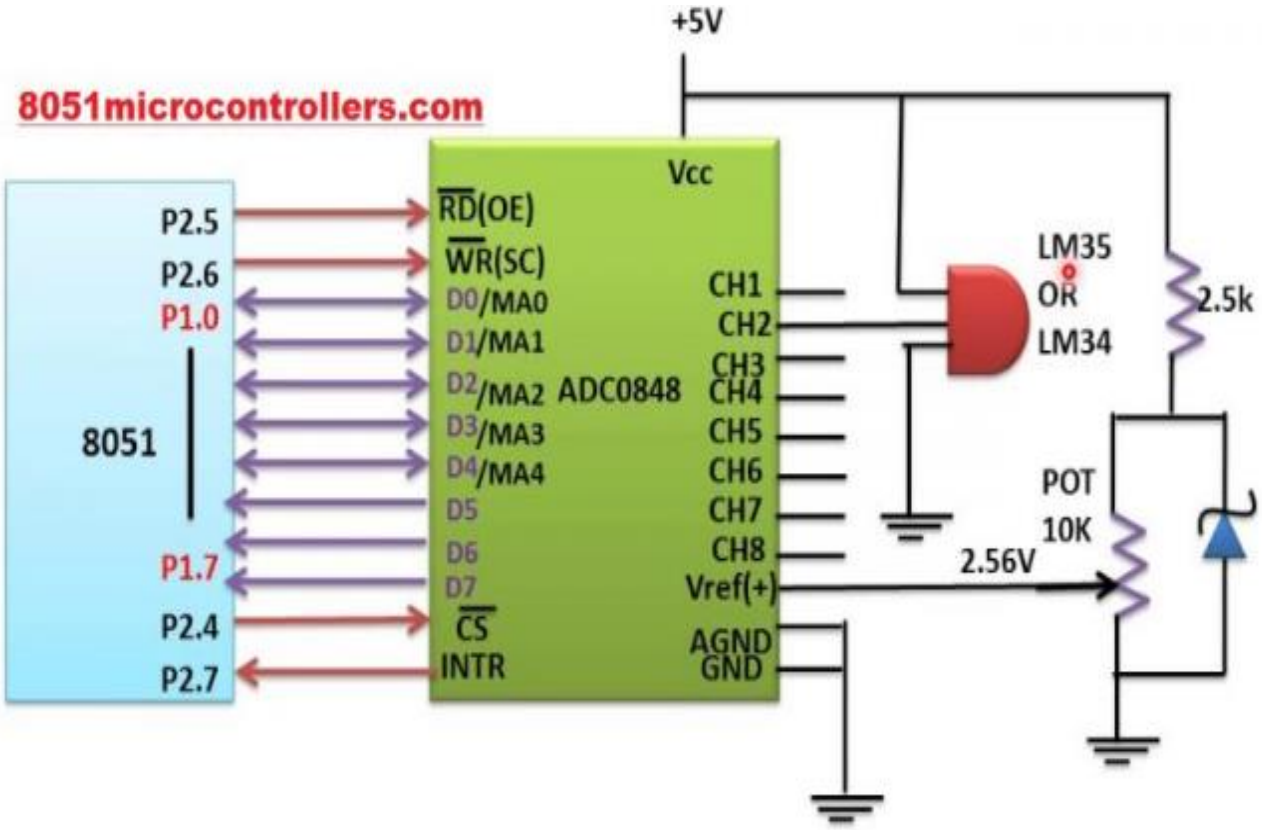


Fig: Interfacing of Lm35 with PIC18f458 for temperature measurement.

Conclusion: LM35 sensor is used for temperature measurement and it is interface with PIC18F458 Here also studied about ADC feature in that different control registers.