SUMMARY

First of all, we will start by understanding the what ADC Module is. ADC is a component that converting the analog signal to the digital signal. Why do we need the ADC? Analog signals change over time and are continuous. Sound, temperature or light levels are examples of analog signals. In order for these signals to be processed in digital systems, they must be converted into digital signals. ADC performs this conversion.

In this project we will understand how an ADC module basically works. We will configure it, write the assembly code and finally do some tests.

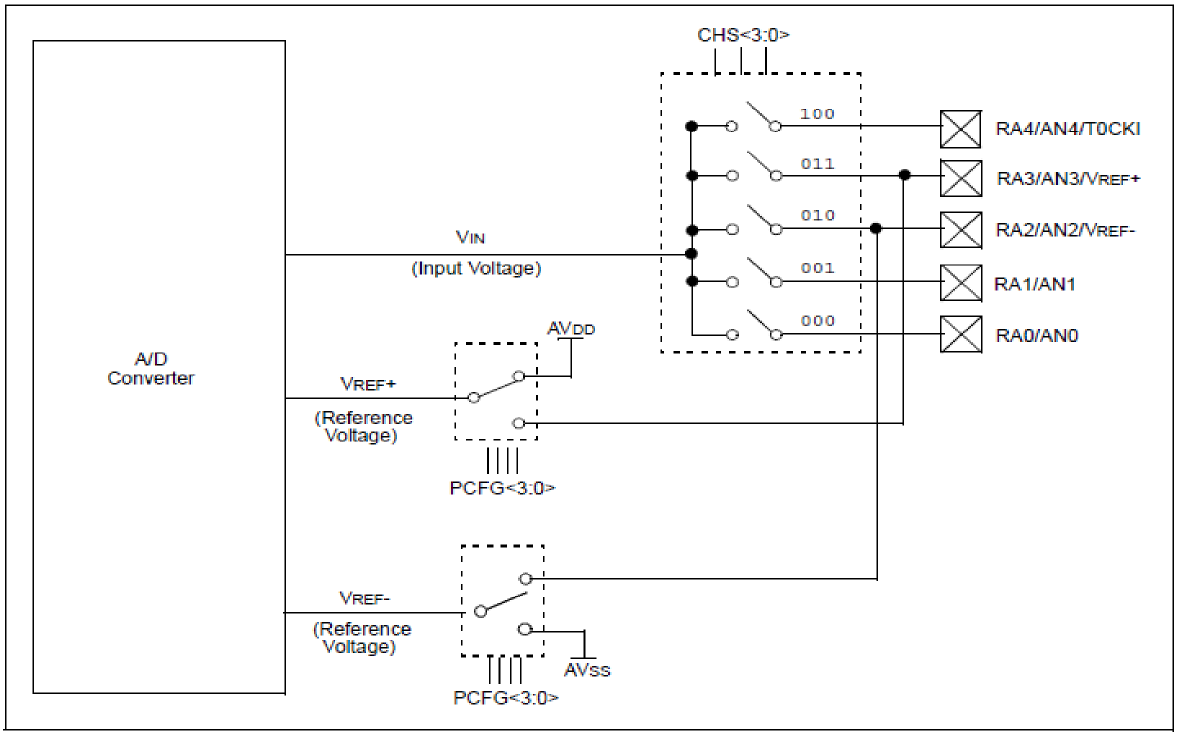
PURPOSE

We are using the ADC in our lives mostly. For example, I'm interested in music and we are taking records with microphone and transfering to computer app. This transferring is made by ADC or, we need to make measurements and this is used by ADC. For instance, measuring the motor speed using some sensors.
This project is going to help the people about converting the analog signal to the digital signal.

INTRODUCTION

First of all, we need to select a microcontroller to create and test the ADC.
I selected the PIC 16F818.

FIGURE 11-1: A/D BLOCK DIAGRAM



-PIC 16F818 has up to five pins, capable of ADC. Those pins are RA0, RA1, RA2, RA3, RA4. These pins can be used as analog pins for ADC purposes.
For our project, we just need one analog input. So, I'm going to use RA0 as the analog input.

-These channels can be selected using the three least significiant bits of our channel select register (CHS <3:0>). For this selection we will use the ADCON0.

REGISTER 11-1: ADCON0: A/D CONTROL REGISTER 0 (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
<u>If ADCS2 = 0:</u>
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from the internal A/D module RC oscillator)
<u>If ADCS2 = 1:</u>
00 = FOSC/4
01 = FOSC/16
10 = FOSC/64
11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
000 = Channel 0 (RA0/AN0)
001 = Channel 1 (RA1/AN1)
010 = Channel 2 (RA2/AN2)
011 = Channel 3 (RA3/AN3)
100 = Channel 4 (RA4/AN4)

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
<u>If ADON = 1:</u>
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

-To perform successfully the ADC, we need positive and negative reference. Positive: AV_{DD} , negative: AV_{SS}.

REGISTER 11-2: ADCON1: A/D CONTROL REGISTER 1 (ADDRESS 9Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified, 6 Most Significant bits of ADRESH are read as '0'
0 = Left justified, 6 Least Significant bits of ADRESL are read as '0'

bit 6 **ADCS2:** A/D Clock Divide by 2 Select bit
1 = A/D clock source is divided by 2 when system clock is used
0 = Disabled

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG<3:0>:** A/D Port Configuration Control bits

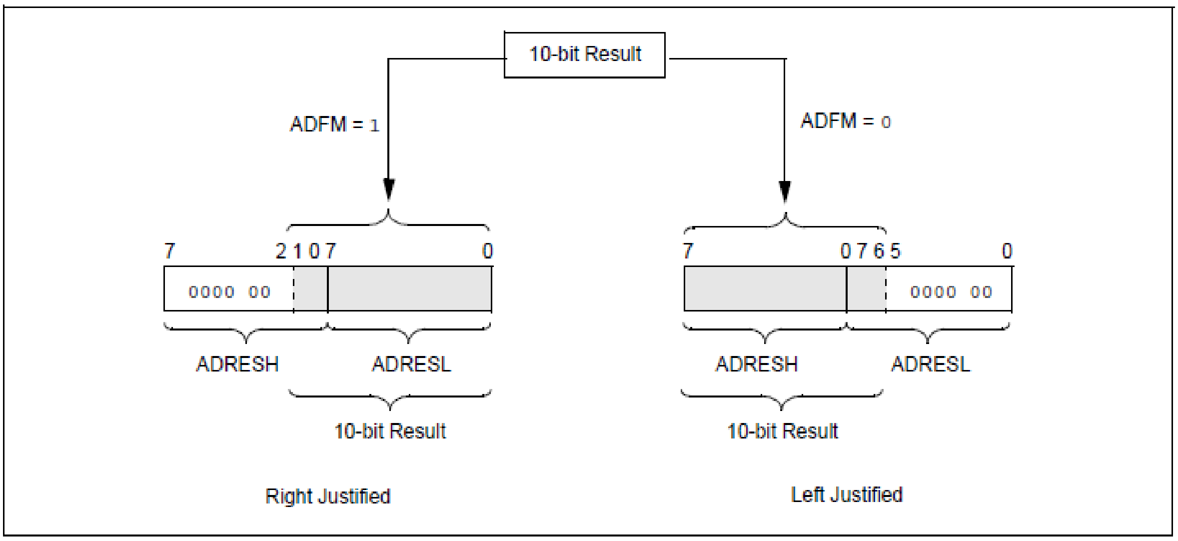
PCFG	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	AVDD	AVSS	5/0
0001	A	VREF+	A	A	A	AN3	AVSS	4/1
0010	A	A	A	A	A	AVDD	AVSS	5/0
0011	A	VREF+	A	A	A	AN3	AVSS	4/1
0100	D	A	D	A	A	AVDD	AVSS	3/0
0101	D	VREF+	D	A	A	AN3	AVSS	2/1
011x	D	D	D	D	D	AVDD	AVSS	0/0
1000	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1001	A	A	A	A	A	AVDD	AVSS	5/0
1010	A	VREF+	A	A	A	AN3	AVSS	4/1
1011	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1100	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	A	AVDD	AVSS	1/0
1111	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = Number of analog input channels/Number of A/D voltage references

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

-We need 1 analog, 4 digitak pins. So, we are going to use 1110. When we perform the ADC, the result that we get is 10 bits but our microcontroller can use maximum 8 bits. How will we handle this problem? Using 2 register. First one will hold 8 bits and the other remaining.

FIGURE 11-4: A/D RESULT JUSTIFICATION



Now, we can start writing our program

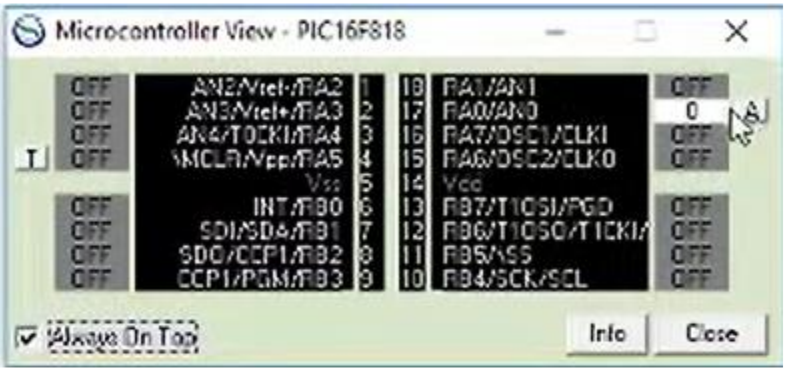
```
1        LIST              P=16F818
2      #INCLUDE       <P16F818.INC>

3        ORG              0X00                  ; reset vector
4        GOTO            INITIALIZE

5    INITIALIZE   BSF                    STATUS, RP0    ; switch to bank 1
6      MOVLW                      0X01
7      MOVWF                      TRISA               ; make only RA0 as input
8      CLRF              TRISB
9      MOVLW                      OX8E
10     MOVWF                      ADCON1           ; AN0 as analog input, VDD and VSS as ADC
reference
11     BCF                 STATUS, RP0
12     MOVLW                      OXC1
13     MOVWF                      ADCON0           ; enable adc module, select channel 0

14 MAIN           BSF            ADCON0, GO     ; initiate adc
15 ADCLOOP    BTFSC                  ADCON0, GO
16                  GOTO                ADCLOOP          ; wait for the ADC to complete
17                 RLF              ADRESH, W  ; read ADRESH ADC results
18                 MOVWF        PORTA                    ; write to RA1 and RA2
19                 BSF              STATUS, RP0           ; bank 1
20                 MOVF           ADRESL, W  ; read ADC results in ADRESL register
21                 BCF              STATUS, RP0           ; bank 0
22                 MOVWF        PORTB                    ; write the value to PORTB
23                 GOTO           MAIN
24
25                 END
```

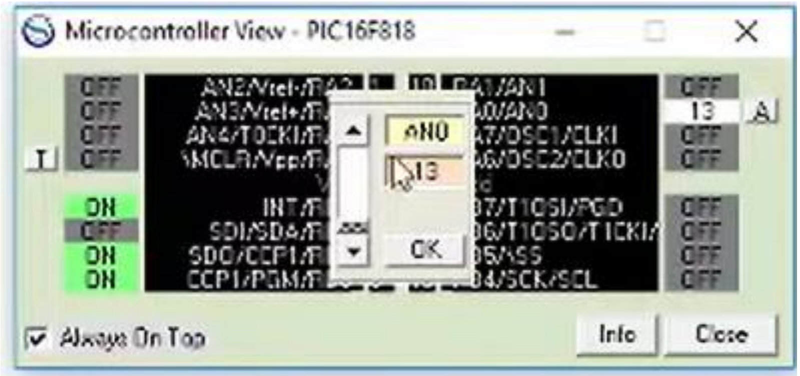
At line 1 and 2, we are selecting the model of microcontroller and adding the file.
At line 3, we are determining the starting address.
At line 4, this program is going to INITIALIZE after resetting.
At line 5, we want to go and set up ADCON1 register. We need to make sure that RA0 as an input. With BSF (bit set file), we are making the flag to 1. When RP0 is 1, bank is 1.
At line 6, we are loading the 0x01 to the W (working register).
At line 7, we are copying the value in the W to the TRISA. TRISA controls the i/o management of PORTA.
At line 8, we reset the TRISB.
At line 9, 0x8E to W.
At line 10, W to ADCON1
At line 11, we reset the RP0 in the STATUS (reset means sets to 0)
At line 12, OXC1 to W.
At line 13, W to ADCON0.
At line 14, GO bits to 1 and operation starts.
At line 15-16, waiting until GO is 0.
At line 17, ADRESH loads to W.
At line 18, write to PORTA
At line 19, to Bank 1.
At line 20, ADRESL to the W.
At line 21, go back to Bank 0

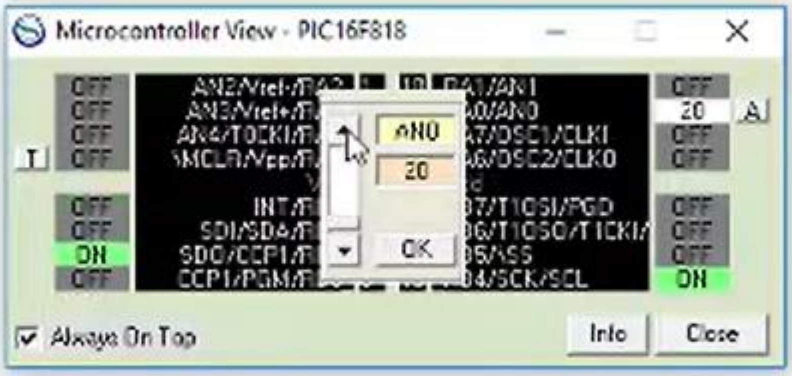
OBSTACLES

I started by researching what adc is. After learning this, I learned that I needed to write the assembly code I needed to write to create ADC on a microcontroller and I chose this microcontroller. I installed the required IDE, but I couldn't run even a simple code. In my first problem, the microcontroller didn’t understand whether the code was C or Assembly. Because although xc8 compiler and xc8 linker were installed, xc8 assembler was not installed. I solved this problem. In the other case, the microcontroller insisted on not recognizing the MAIN function, which is the main function, but I solved this problem as well.

SOME TESTS AND RESULTS

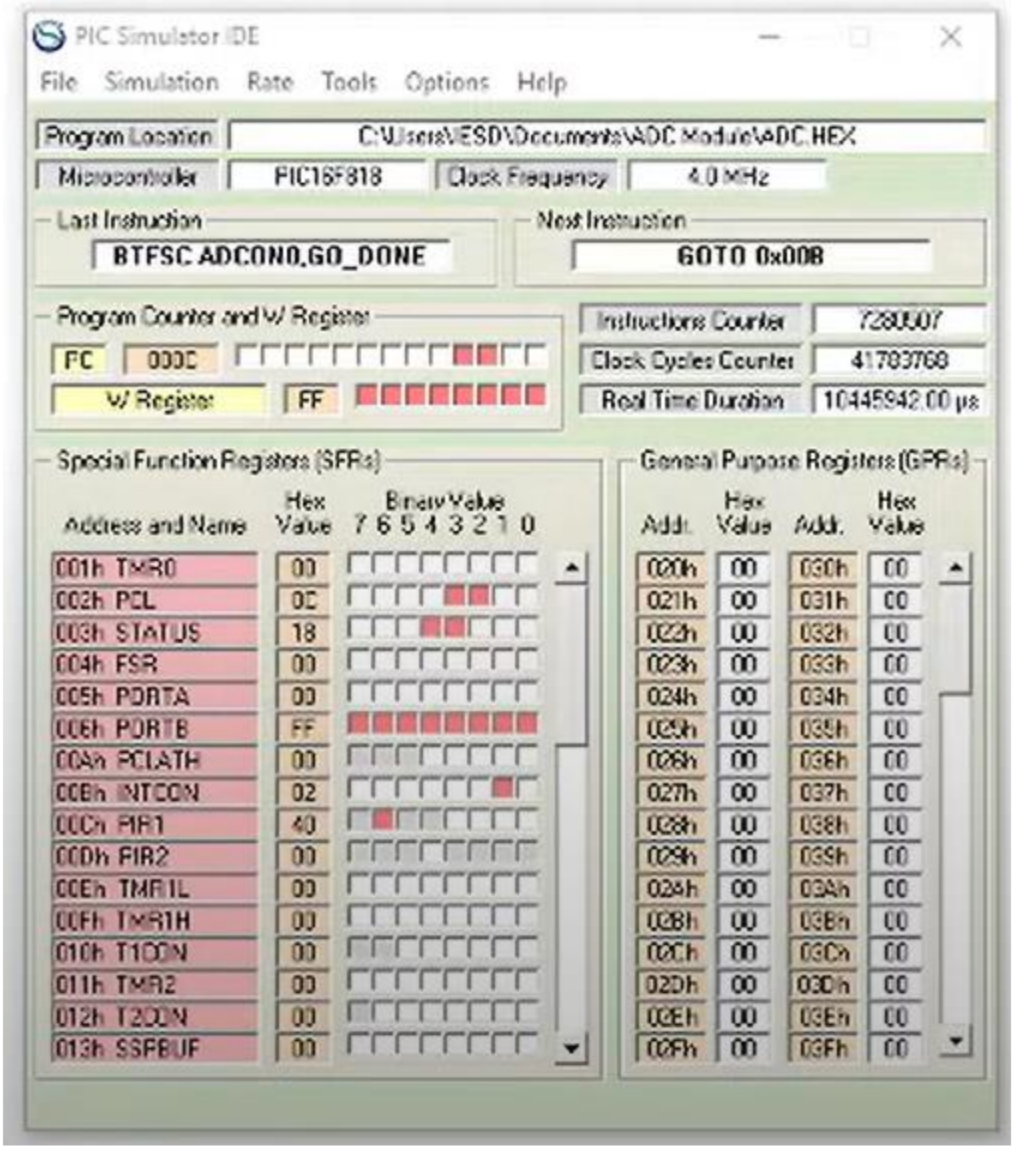


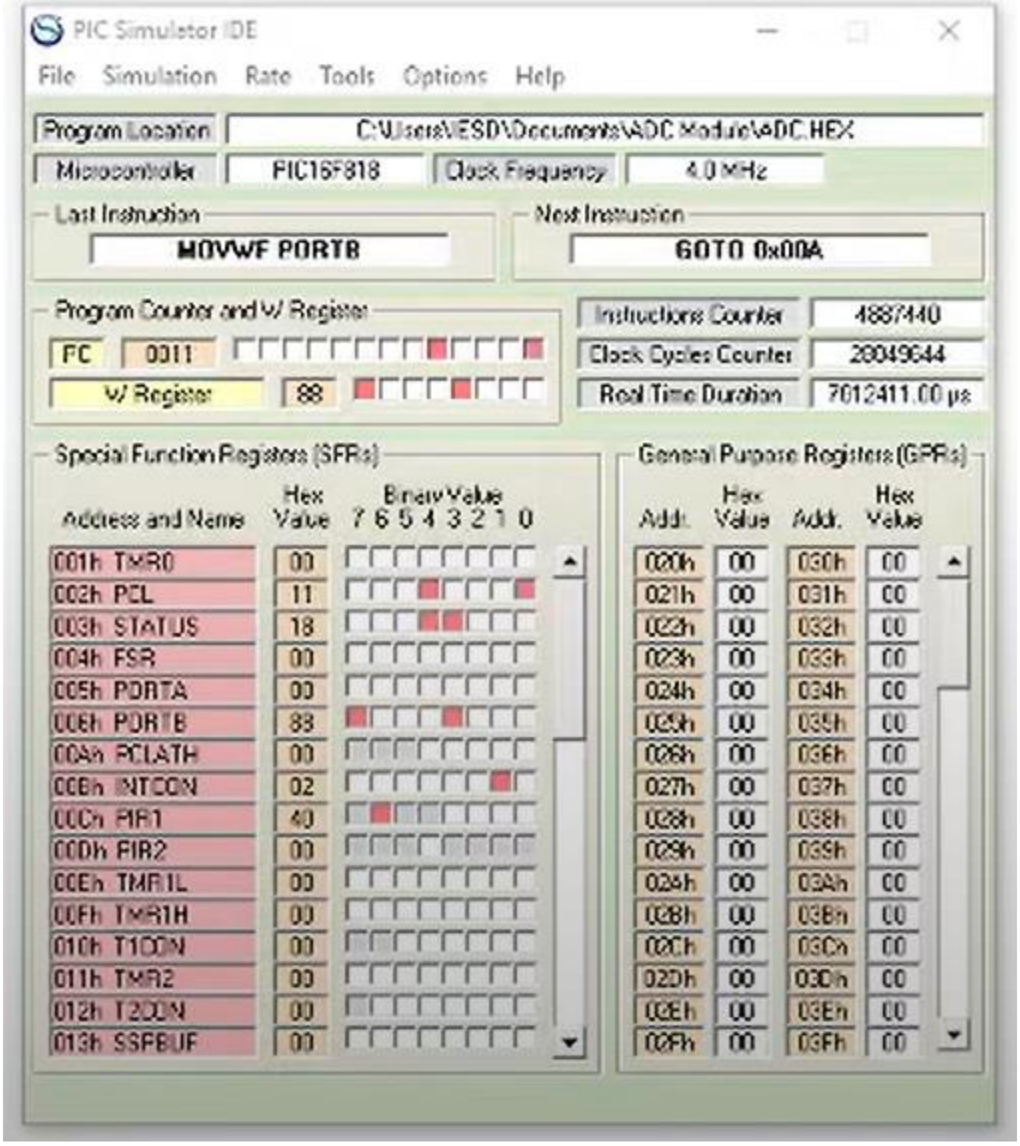


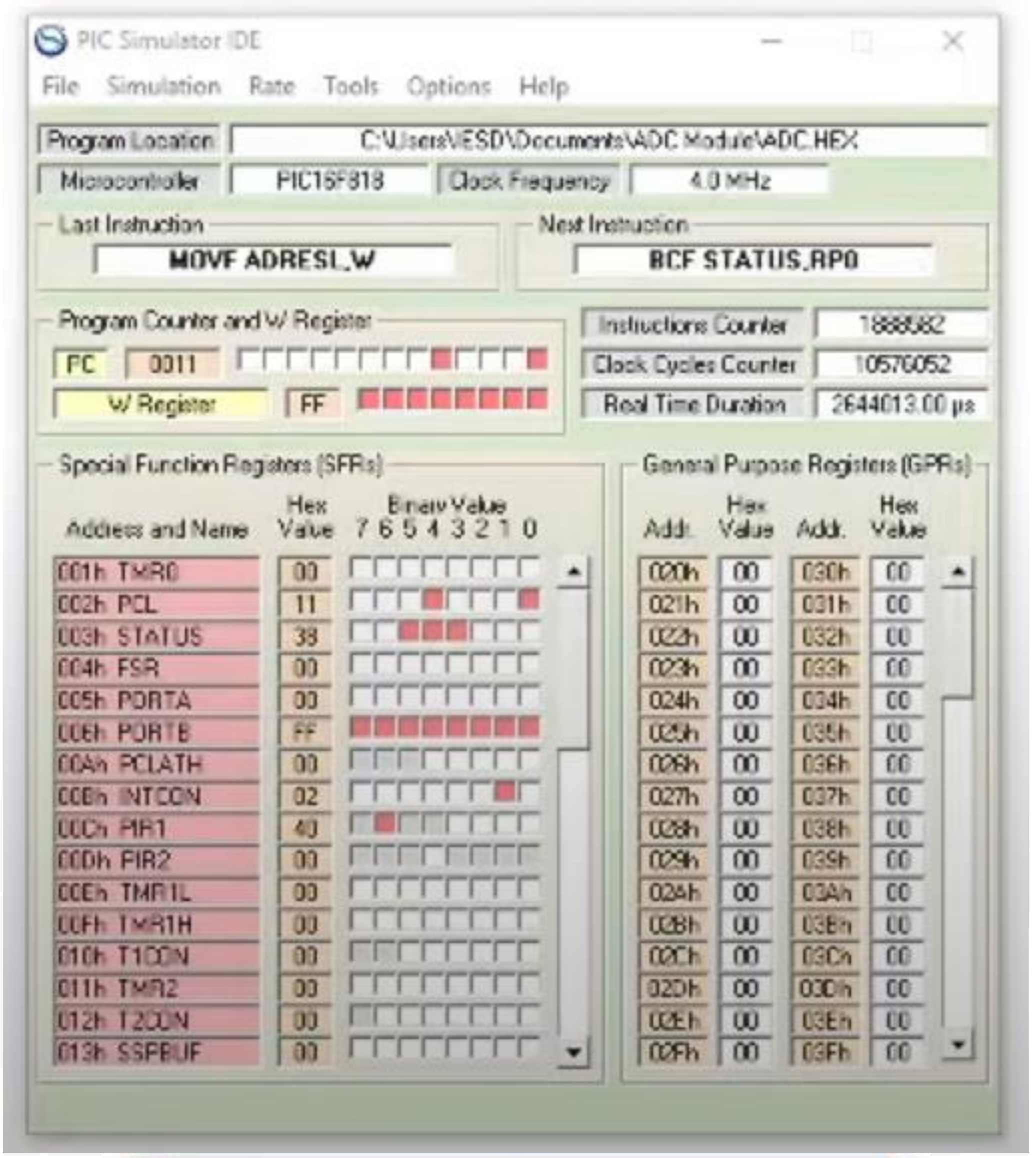

Microcontroller View - PIC16F818
AN0
20
OK
Always On Top
Info
Close


Microcontroller View - PIC16F818
AN0
111
OK
Always On Top
Info
Close

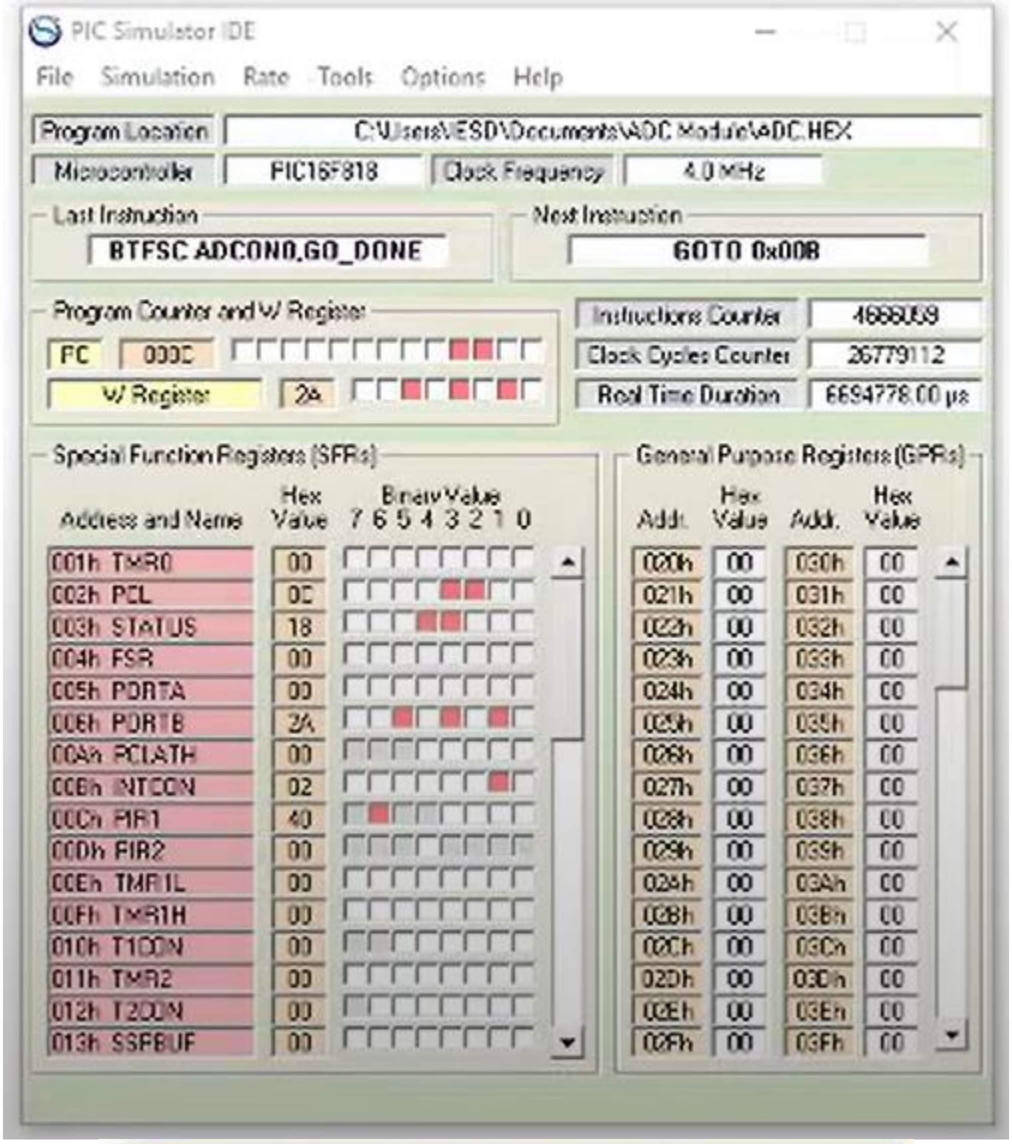

Microcontroller View - PIC16F818
AN0
255
OK
Always On Top
Info
Close


PIC Simulator IDE
File
Simulation
Rate
Tools
Options
Help
Program Location
Microcontroller
PIC16F818
Clock Frequency
4.0 MHz
Last Instruction
BTFSC ADCON0,GO_DONE
Next Instruction
GOTO 0x00B
Program Counter and W Register
PC
000C
W Register
FF
Instructions Counter
Clock Cycles Counter
41783768
Real Time Duration
Special Function Registers (SFRs)
Address and Name
Hex Value
Binary Value
7 6 5 4 3 2 1 0
001h TMR0
002h PCL
003h STATUS
004h FSR
005h PORTA
006h PORTB
00Ah PCLATH
00Bh INTCON
00Ch PIR1
00Dh PIR2
00Eh TMR1L
00Fh TMR1H
010h T1CON
011h TMR2
012h T2CON
013h SSPBUF
General Purpose Registers (GPRs)
Addr.
Hex Value

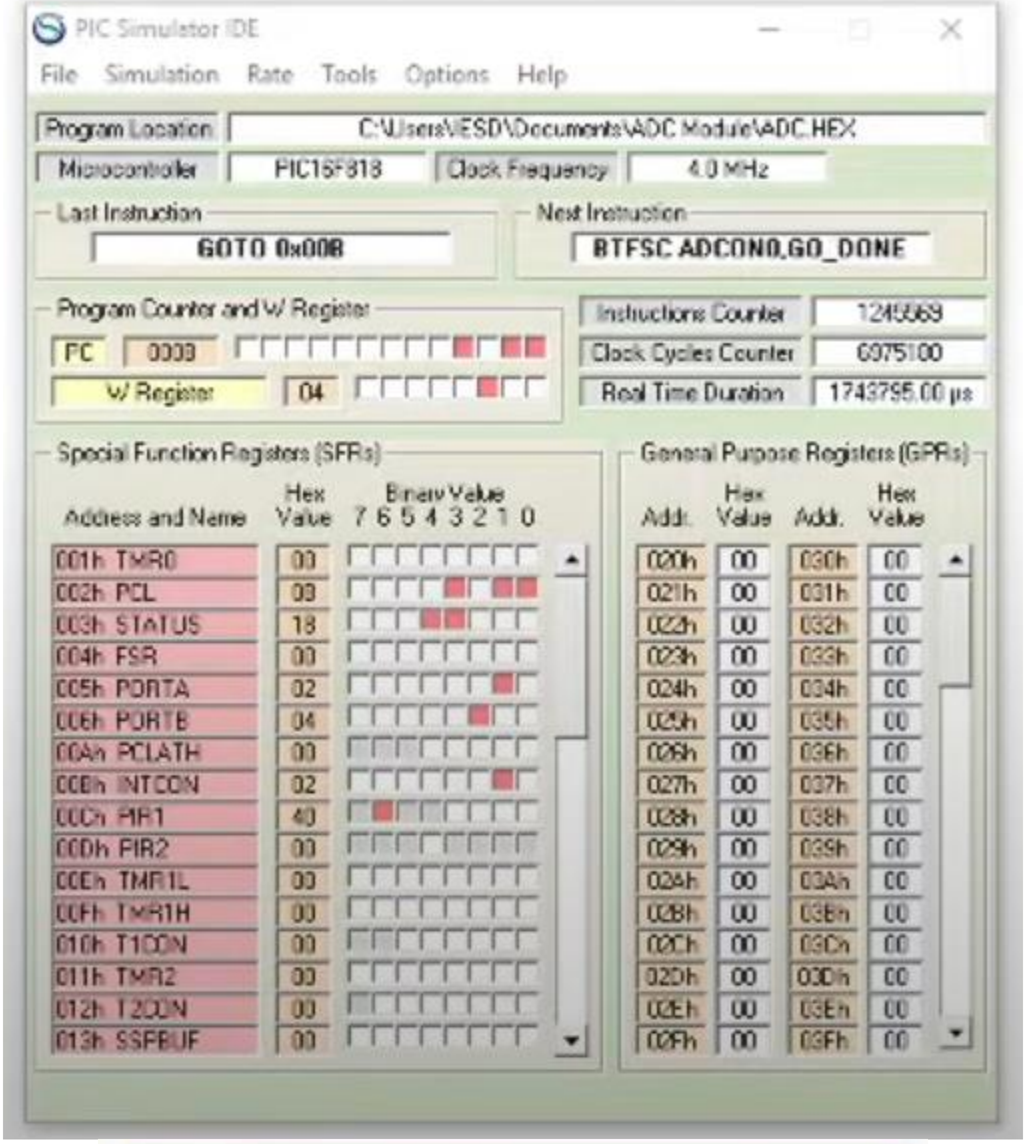

PIC Simulator IDE
File Simulation Rate Tools Options Help
Program Location
Microcontroller
PIC16F818
Clock Frequency
4.0 MHz
Last Instruction
MOVWF PORTB
Next Instruction
GOTO 0x00A
Program Counter and W Register
PC
0011
W Register
88
Instructions Counter
4887440
Clock Cycles Counter
Real Time Duration
Special Function Registers (SFRs)
Address and Name
Hex Value
Binary Value
7 6 5 4 3 2 1 0
001h TMR0
002h PCL
003h STATUS
004h FSR
005h PORTA
006h PORTB
00Ah PCLATH
00Bh INTCON
00Ch PIR1
00Dh PIR2
00Eh TMR1L
00Fh TMR1H
010h T1CON
011h TMR2
012h T2CON
013h SSPBUF
General Purpose Registers (GPRs)
Addr.
Hex Value


PIC Simulator IDE
File Simulation Rate Tools Options Help
Program Location
C:\Users\ESD\Documents\ADC Module\ADC.HEX
Microcontroller
PIC16F818
Clock Frequency
4.0 MHz
Last Instruction
MOVF ADRESL,W
Next Instruction
BCF STATUS,RP0
Program Counter and W Register
PC
0011
W Register
FF
Instructions Counter
1888582
Clock Cycles Counter
10576052
Real Time Duration
2644013.00 µs
Special Function Registers (SFRs)
Address and Name
Hex Value
Binary Value
7 6 5 4 3 2 1 0
001h TMR0 00
002h PCL 11
003h STATUS 38
004h FSR 00
005h PORTA 00
006h PORTB FF
00Ah PCLATH 00
00Bh INTCON 02
00Ch PIR1 40
00Dh PIR2 00
00Eh TMR1L 00
00Fh TMR1H 00
010h T1CON 00
011h TMR2 00
012h T2CON 00
013h SSPBUF 00
General Purpose Registers (GPRs)
Addr. Hex Value Addr. Hex Value
020h 00 030h 00
021h 00 031h 00
022h 00 032h 00
023h 00 033h 00
024h 00 034h 00
025h 00 035h 00
026h 00 036h 00
027h 00 037h 00
028h 00 038h 00
029h 00 039h 00
02Ah 00 03Ah 00
02Bh 00 03Bh 00
02Ch 00 03Ch 00
02Dh 00 03Dh 00
02Eh 00 03Eh 00
02Fh 00 03Fh 00

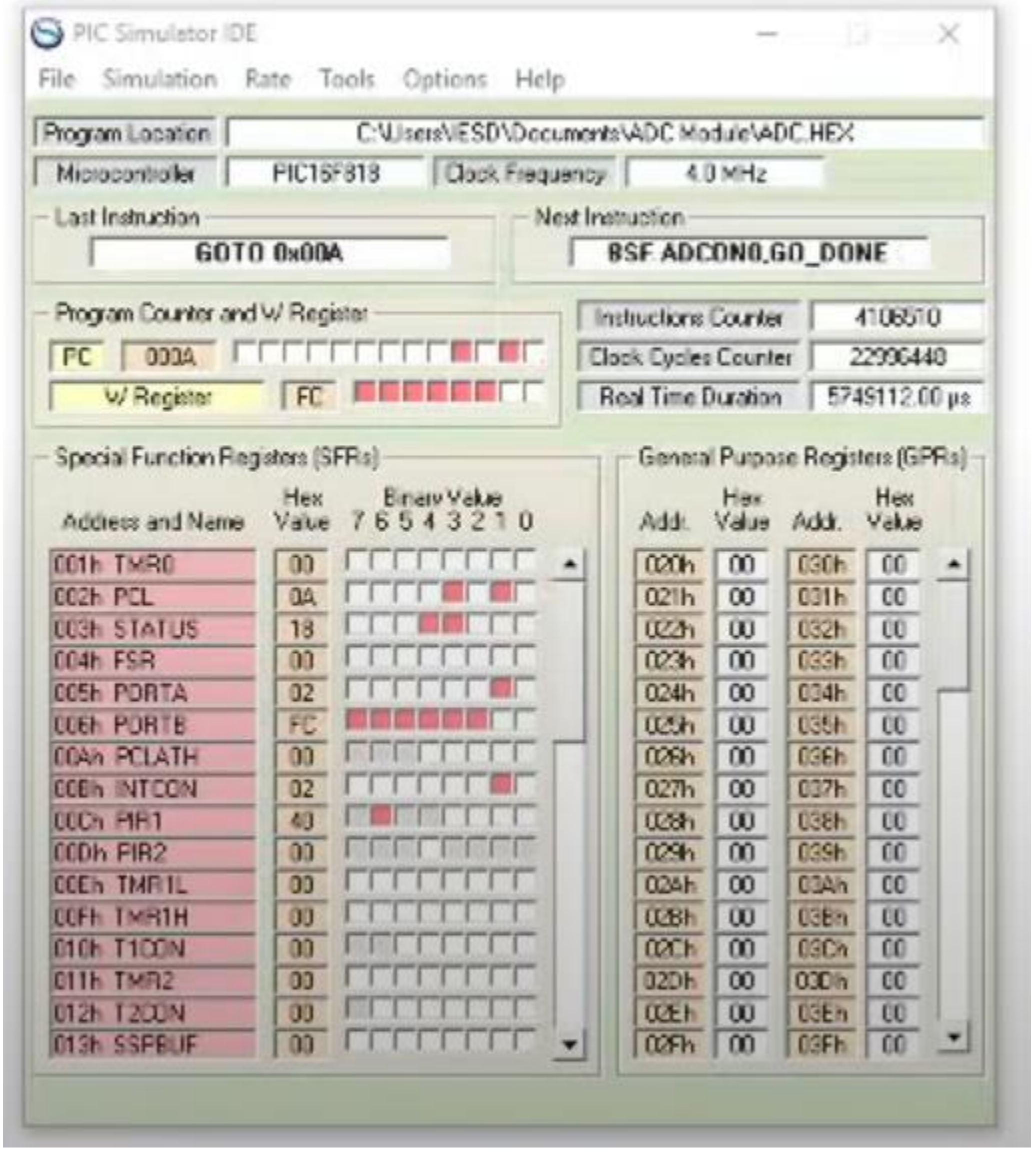

Microcontroller View - PIC16F818
OFF
OFF
OFF
OFF
T
OFF
OFF
ON
ON
AN0
509
OK
ON
509
A
OFF
OFF
ON
ON
ON
ON
Always On Top
Info
Close


PIC Simulator IDE
File Simulation Rate Tools Options Help
Program Location
C:\Users\VESD\Documents\ADC Module\ADC.HEX
Microcontroller
PIC16F818
Clock Frequency
4.0 MHz
Last Instruction
BTFSC ADCON0,GO_DONE
Next Instruction
GOTO 0x00B
Program Counter and W Register
PC
000C
W Register
2A
Instructions Counter
Clock Cycles Counter
26779112
Real Time Duration
Special Function Registers (SFRs)
Address and Name
Hex Value
Binary Value
7 6 5 4 3 2 1 0
001h TMR0
002h PCL
003h STATUS
004h FSR
005h PORTA
006h PORTB
00Ah PCLATH
00Bh INTCON
00Ch PIR1
00Dh PIR2
00Eh TMR1L
00Fh TMR1H
010h T1CON
011h TMR2
012h T2CON
013h SSPBUF
General Purpose Registers (GPRs)
Addr.
Hex Value


Microcontroller View - PIC16F818
AN0
260
OK
Always On Top
Info
Close


PIC Simulator IDE
File Simulation Rate Tools Options Help
Program Location
C:\Users\ESD\Documents\ADC Module\ADC.HEX
Microcontroller
PIC16F818
Clock Frequency
4.0 MHz
Last Instruction
GOTO 0x008
Next Instruction
BTFSC ADCON0,GO_DONE
Program Counter and W Register
PC
0008
W Register
04
Instructions Counter
1245569
Clock Cycles Counter
6975100
Real Time Duration
1743795.00 µs
Special Function Registers (SFRs)
Address and Name
Hex Value
Binary Value
7 6 5 4 3 2 1 0
001h TMR0 00
002h PCL 08
003h STATUS 18
004h FSR 00
005h PORTA 02
006h PORTB 04
00Ah PCLATH 00
00Bh INTCON 02
00Ch PIR1 40
00Dh PIR2 00
00Eh TMR1L 00
00Fh TMR1H 00
010h T1CON 00
011h TMR2 00
012h T2CON 00
013h SSPBUF 00
General Purpose Registers (GPRs)
Addr. Hex Value Addr. Hex Value
020h 00 030h 00
021h 00 031h 00
022h 00 032h 00
023h 00 033h 00
024h 00 034h 00
025h 00 035h 00
026h 00 036h 00
027h 00 037h 00
028h 00 038h 00
029h 00 039h 00
02Ah 00 03Ah 00
02Bh 00 03Bh 00
02Ch 00 03Ch 00
02Dh 00 03Dh 00
02Eh 00 03Eh 00
02Fh 00 03Fh 00


Microcontroller View - PIC16F818
OFF
OFF
OFF
OFF
ON
ON
ON
ON
AN0
255
OK
255
A
T
Always On Top
Info
Close


PIC Simulator IDE
File Simulation Rate Tools Options Help
Program Location
C:\Users\ESD\Documents\ADC Module\ADC.HEX
Microcontroller
PIC16F818
Clock Frequency
4.0 MHz
Last Instruction
GOTO 0x00A
Next Instruction
BSF ADCON0,GO_DONE
Program Counter and W Register
PC
000A
W Register
FC
Instructions Counter
4106510
Clock Cycles Counter
22996448
Real Time Duration
5749112.00 µs
Special Function Registers (SFRs)
Address and Name
Hex Value
Binary Value
7 6 5 4 3 2 1 0
001h TMR0 00
002h PCL 0A
003h STATUS 18
004h FSR 00
005h PORTA 02
006h PORTB FC
00Ah PCLATH 00
00Bh INTCON 02
00Ch PIR1 40
00Dh PIR2 00
00Eh TMR1L 00
00Fh TMR1H 00
010h T1CON 00
011h TMR2 00
012h T2CON 00
013h SSPBUF 00
General Purpose Registers (GPRs)
Addr. Hex Value Addr. Hex Value
020h 00 030h 00
021h 00 031h 00
022h 00 032h 00
023h 00 033h 00
024h 00 034h 00
025h 00 035h 00
026h 00 036h 00
027h 00 037h 00
028h 00 038h 00
029h 00 039h 00
02Ah 00 03Ah 00
02Bh 00 03Bh 00
02Ch 00 03Ch 00
02Dh 00 03Dh 00
02Eh 00 03Eh 00
02Fh 00 03Fh 00

RESOURCES
-PIC 16F818 DataSheet
https://www.alldatasheet.com/view.jsp?Searchword=Pic16f818%20datasheet&gad_source=1&gclid=Cj0KCQiA3sq6BhD2ARIsAJ8MRwVGWFqJGntcBm6ybRPsQud1D-FmLFwnhMHeCIm2cebkQFiHb6JRgk4aAsZSEALw_wcB

- Assembly Programming for PIC Microcontroller Embedded Systems by Charles kim
https://www.mwftr.com/book/PIC_CharlesKim_08.pdf

- https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html