Internet Clock using PIC16F887

Ashwin Vallaban

Fr. Conceicao Rodrigues Institute of Technology, Vashi

Feb 2023

Table of Contents

Software Used

Software	Version	Used for
MPLAB X IDE	6.05	Configuration Bits
mikroC Pro	7.6.0	PIC Programming
QL-PROG for QL-2006	2.37	PIC Burner
Proteus Design Suite	8.9 SP2	Simulation
Arduino IDE	1.8.19	ESP-01 Programming with CP2102 UART Serial Converter

Table of Figures

Table of Code Snippets

1 Introduction and Summary of the Project

Internet Clock is a project which shows time in HH: MM. Although the firmware is also compatible with all the series of PIC16F88x, PIC16F887 is primarily described in this documentation. A Real Time Clock (RTC) IC is used to keep the track of time. Tactile switches are provided which can also be used to adjust the time manually. Clock can also be synchronized with the internet time using a Wi-Fi module (ESP8266-01). The time is displayed on 7 segment displays driven by a display driver. A Potentiometer is used to set the brightness of the 7 segment displays using the measured ADC value.

Peripherals used:

- **I2C** between PIC and RTC
- **UART** between PIC and Wi-Fi module
- **SPI** between PIC and 7 segment display driver
- **ADC** for measuring the POT value
- **Timer/counter** in PIC for keeping track of seconds elapsed

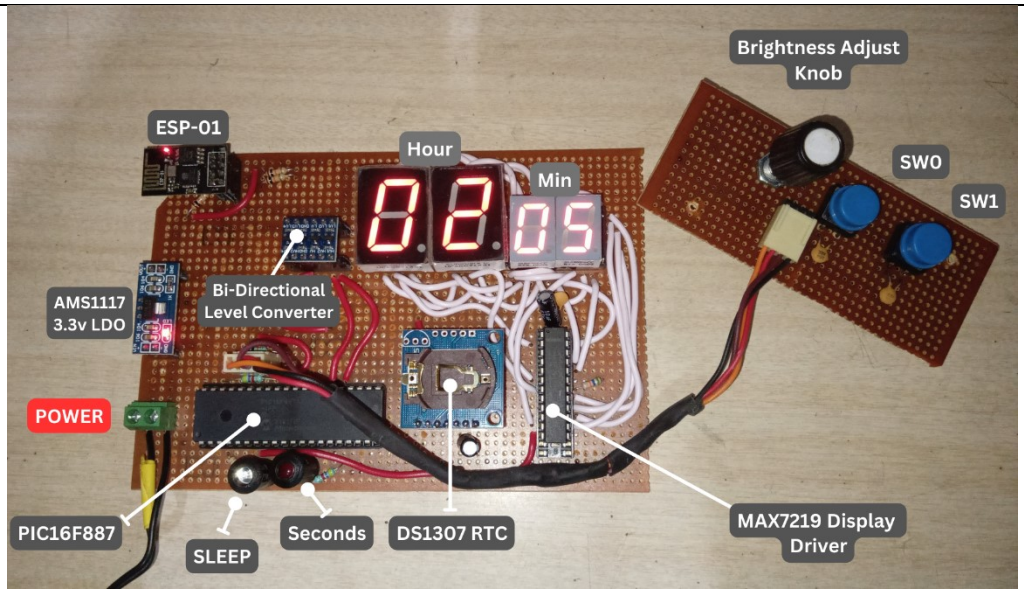


Figure 1 General Overview of the Project

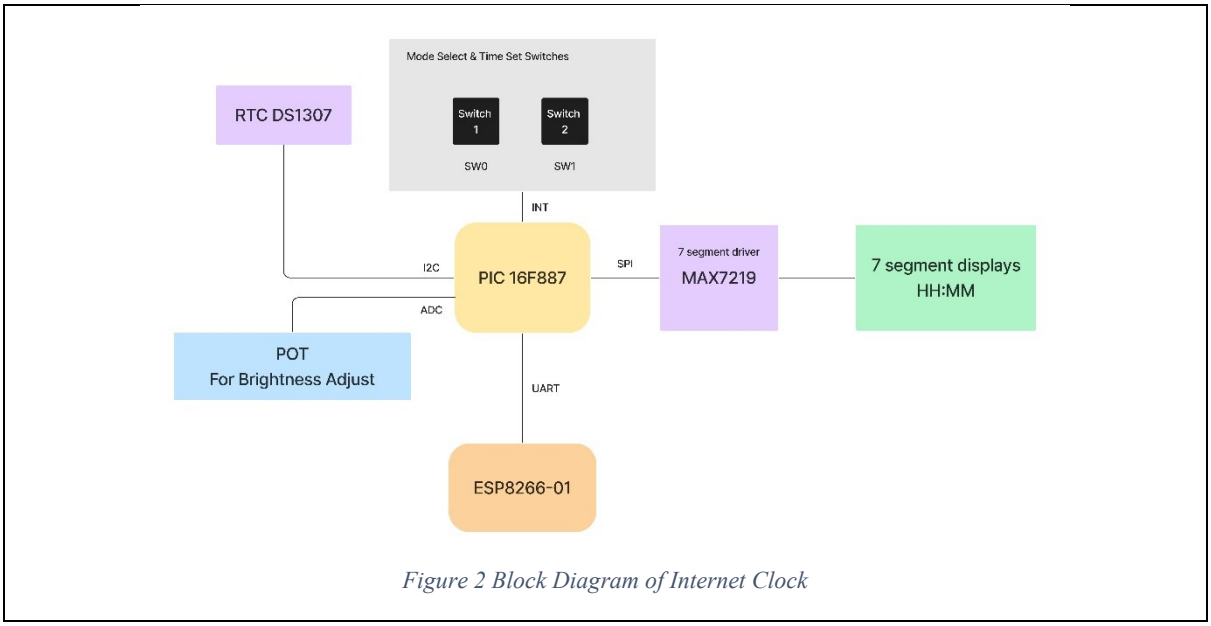


Figure 2 Block Diagram of Internet Clock

Figure 2 shows the block diagram of the Internet Clock Project. The overall simulation in Proteus is shown in figure 3.

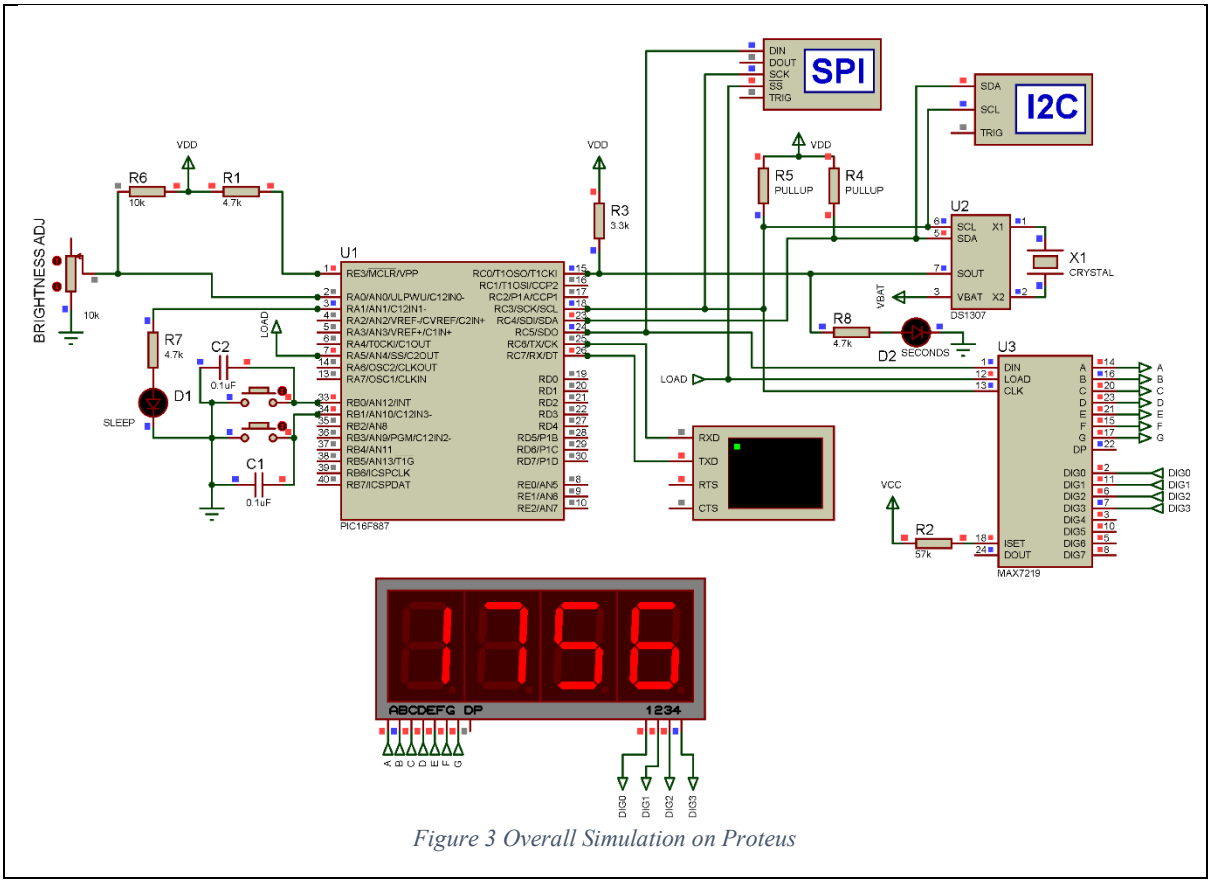


Figure 3 Overall Simulation on Proteus

2 Design Consideration

2.1 Introduction to PIC16F887

2.1.1 Memory Organization

PIC16F88x series microcontroller is offered by Microchip. As quoted in their datasheet, it is a 28/40/44-Pin, Enhanced Flash-Based 8-Bit CMOS Microcontrollers with nanoWatt Technology. This project uses PIC16F887 which is a 8-bit midrange microcontroller. Some of the features which lead to using this microcontroller for the project was:

- Internal Precision Oscillator with range 8MHz to 31kHz which can be tuned with software
- Power saving sleep mode
- Power On Reset (POR)
- Oscillator Start-up Timer (OST)
- 8K Bytes Program Memory (Flash Word of 14 bit)
- 368 Bytes SRAM
- 10-bit ADC
- 8-bit and 16-bit Timer/Counter
- Master Synchronous Serial Port (MSSP) supporting I2C and SPI
- Enhanced USART

It has a 13-bit Program Counter (PC) addressing from 0000h to 1FFFh. Reset vector is at 0000h and Interrupt vector is at 0004h.

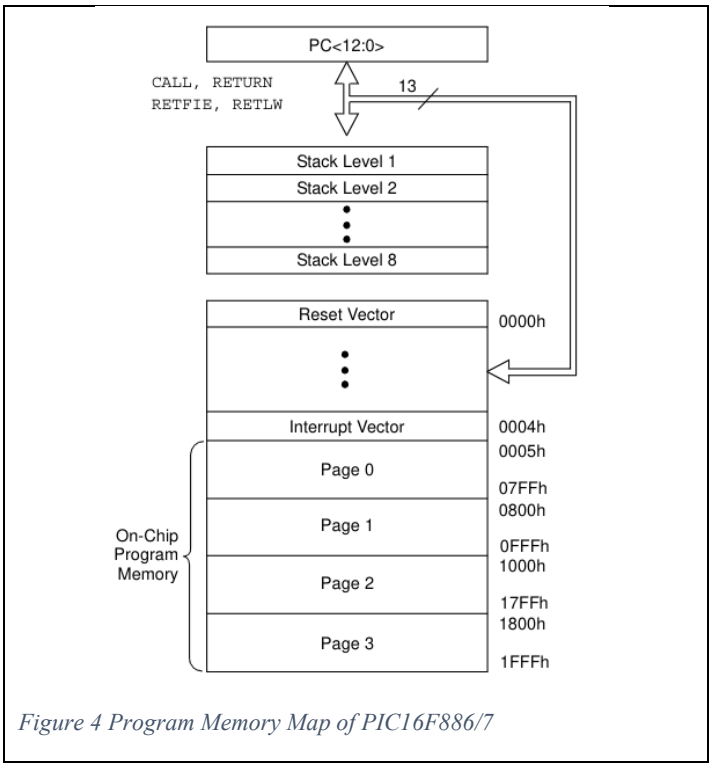


Figure 4 Program Memory Map of PIC16F886/7

Data memory is partitioned into 4 banks ranging from 000h to 1FFh. It consists of General-Purpose Registers (GPR) and Special Function Registers (SFR). The SFR is located in the first 32 locations of each bank. GPR are located in the last 96 locations of each bank. In order to select a bank, RP1 & RP0 bits in STATUS register <03h> is used.

	File Address		File Address		File Address		File Address
Indirect addr. (1)	00h	Indirect addr. (1)	80h	Indirect addr. (1)	100h	Indirect addr. (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	WDTCON	105h	SRCON	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	CM1CON0	107h	BAUDCTL	187h
PORTD(2)	08h	TRISD(2)	88h	CM2CON0	108h	ANSEL	188h
PORTE	09h	TRISE	89h	CM2CON1	109h	ANSELH	189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved	18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh	Reserved	18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUB	95h		115h		195h
CCPR1H	16h	IOCB	96h	General Purpose Registers 16 Bytes	116h	General Purpose Registers 16 Bytes	196h
CCP1CON	17h	VRCON	97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah	SPBRGH	9Ah		11Ah		19Ah
CCPR2L	1Bh	PWM1CON	9Bh		11Bh		19Bh
CCPR2H	1Ch	ECCPAS	9Ch		11Ch		19Ch
CCP2CON	1Dh	PSTRCON	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Registers 96 Bytes	20h 3Fh 40h 6Fh	General Purpose Registers 80 Bytes	A0h EFh	General Purpose Registers 80 Bytes	120h 16Fh	General Purpose Registers 80 Bytes	1A0h 1EFh
	70h 7Fh	accesses 70h-7Fh	F0h FFh	accesses 70h-7Fh	170h 17Fh	accesses 70h-7Fh	1F0h 1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

Figure 5 Special Function Registers

2.1.2 IO Ports

IO pins may have different functionalities other than general purpose input output depending on the peripherals which are enabled. PIC16F887 consists of PORTA, PORTB, PORTC, PORTD & PORTE. Corresponding TRISx register is used to indicate if the PORTx pin will function as an input or an output. If TRISx = 1, the pin acts as an input and when TRISx = 0, the pin acts as an output. Configuring as an input disables the output driver driver and vice-versa. Writing to a port indicated writing to a latch which read operation reads the status of the port pin. All operations are Read-Modify-Write. Some port pins may be used as ADC input. ANSEL register <188h> is used to configure an Analog pin to be used as Digital I/O. Setting (=1) ANSEL register bit ANSx where x is the Analog Port selected makes the pin Analog while Clearing (=0) makes the pin Digital. Default state on Reset if Analog for all the compatible port pins.

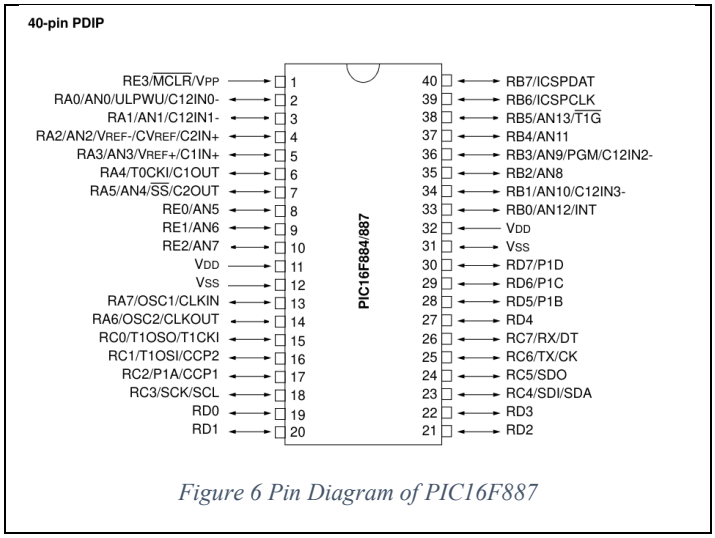


Figure 6 Pin Diagram of PIC16F887

2.1.3 Configuration Bits (Fuses), Oscillator Selection and Basic Circuitry

Configuration Bits or Fuses are programmed while burning the firmware onto the microcontroller. These bits are mapped to location 2007h. This location is beyond the user program memory space and can be only accessed while programming the chip. Configuration bits consists of CONFIG1 Register and CONFIG2 registers.

—	—	$\overline{\text{DEBUG}}$	LVP	FCMEN	IESO	BOREN1	BOREN0
bit 15							bit 8

$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7							bit 0

bit 15-14 **Unimplemented**: Read as '1'

bit 13 **$\overline{\text{DEBUG}}$:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6/ICSPCLK and RB7/ICSPDAT are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6/ICSPCLK and RB7/ICSPDAT are dedicated to the debugger

bit 12 **LVP:** Low Voltage Programming Enable bit
1 = RB3/PGM pin has PGM function, low voltage programming enabled
0 = RB3 pin is digital I/O, HV on MCLR must be used for programming

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enabled bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled

bit 10 **IESO:** Internal External Switchover bit
1 = Internal/External Switchover mode is enabled
0 = Internal/External Switchover mode is disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits(1)
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
01 = BOR controlled by SBOREN bit of the PCON register
00 = BOR disabled

bit 7 **$\overline{\text{CPD}}$:** Data Code Protection bit(2)
1 = Data memory code protection is disabled
0 = Data memory code protection is enabled

bit 6 **$\overline{\text{CP}}$:** Code Protection bit(3)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** RE3/MCLR pin function select bit(4)
1 = RE3/MCLR pin function is MCLR
0 = RE3/MCLR pin function is digital input, MCLR internally tied to VDD

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled by SWDTEN bit of the WDTCON register

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
110 = RCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
011 = EC: I/O function on RA6/OSC2/CLKOUT pin, CLKIN on RA7/OSC1/CLKIN
010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

Figure 7 Configuration Register 1

—	—	—	—	—	WRT1	WRT0	BOR4V
bit 15							bit 8

—	—	—	—	—	—	—	—
bit 7							bit 0

bit 15-11 **Unimplemented**: Read as '1'

bit 10-9 **WRT<1:0>:** Flash Program Memory Self Write Enable bits

PIC16F883/PIC16F884

00 = 0000h to 07FFh write protected, 0800h to 0FFFh may be modified by EECON control
01 = 0000h to 03FFh write protected, 0400h to 0FFFh may be modified by EECON control
10 = 0000h to 00FFh write protected, 0100h to 0FFFh may be modified by EECON control
11 = Write protection off

PIC16F886/PIC16F887

00 = 0000h to 0FFFh write protected, 1000h to 1FFFh may be modified by EECON control
01 = 0000h to 07FFh write protected, 0800h to 1FFFh may be modified by EECON control
10 = 0000h to 00FFh write protected, 0100h to 1FFFh may be modified by EECON control
11 = Write protection off

PIC16F882

00 = 0000h to 03FFh write protected, 0400h to 07FFh may be modified by EECON control
01 = 0000h to 00FFh write protected, 0100h to 07FFh may be modified by EECON control
11 = Write protection off

bit 8 **BOR4V**: Brown-out Reset Selection bit

0 = Brown-out Reset set to 2.1V
1 = Brown-out Reset set to 4.0V

bit 7-0 **Unimplemented**: Read as '1'

Figure 8 Configuration Register 2

In the project Internal factory calibrated oscillator is used. The frequency can be varied from 8MHz to 31kHz. OSCCON Register <8Fh> is used to configure the oscillator while clock source is selected by setting the FOSC<2:0> bits in CONFIG1. For this project, 4MHz Internal Oscillator is used. For configuring the oscillator, CONFIG1 FOSC bits are set to *100* which is *INTOSCIO mode* so that I/O functions in RA6 and RA7 pins can be preserved. 4MHz is the default Internal Clock speed in the OSCCON Register <8Fh>. Low Voltage programming is not desired so the LVP bit is cleared.

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS(1)	HTS	LTS	SCS
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits

111 = 8 MHz
110 = 4 MHz (default)
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (LFINTOSC)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit(1)

1 = Device is running from the external clock defined by FOSC<2:0> of the CONFIG1 register
0 = Device is running from the internal oscillator (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC Status bit (High Frequency – 8 MHz to 125 kHz)

1 = HFINTOSC is stable
0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC Stable bit (Low Frequency – 31 kHz)

1 = LFINTOSC is stable
0 = LFINTOSC is not stable

bit 0 **SCS:** System Clock Select bit

1 = Internal oscillator is used for system clock
0 = Clock source defined by FOSC<2:0> of the CONFIG1 register

Figure 9 OSCCON Register

PIC16F887 consists of an On-Chip Power-On Reset (POR). However, it also consists of a Brown-out Reset (BOR) which if enabled keeps the chip in Reset until VDD reached BOR. For the project BOR is enabled by setting BOREN to 1 and in the CONFIG2 register BOR4V is set to 1(BOR is 4V).

MCLR (Active Low) pin is used to Reset the PIC16F887. It has a noise filter which will ignore small pulses. Microchip recommends MCLR should not be connected directly to VDD and a resistor with 1kΩ or greater should be used. Internal MCLR option can be enabled by clearing the MCLRE bit in the CONFIG1 register. In the project MCLRE is set so RA3/MCLR pin becomes external reset with weak pull-up to VDD.

Address	Name	Value	Field	Option	Category	Setting
2007	CONFIG1	2FF4	-	-	-	-
		4	FOSC	INTRC_NOCLKOUT	Oscillator Selection bits	INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function o...
		0	WDTE	OFF	Watchdog Timer Enable bit	WDT disabled and can be enabled by SWDTEN bit of the WDTCON register
		1	PWRTE	OFF	Power-up Timer Enable bit	PWRT disabled
		1	MCLRE	ON	RE3/MCLR pin function select bit	RE3/MCLR pin function is MCLR
		1	CP	OFF	Code Protection bit	Program memory code protection is disabled
		1	CPD	OFF	Data Code Protection bit	Data memory code protection is disabled
		3	BOREN	ON	Brown Out Reset Selection bits	BOR enabled
		1	IESO	ON	Internal External Switchover bit	Internal/External Switchover mode is enabled
		1	FCMEN	ON	Fail-Safe Clock Monitor Enabled bit	Fail-Safe Clock Monitor is enabled
		0	LVP	OFF	Low Voltage Programming Enable bit	RB3 pin has digital I/O, HV on MCLR must be used for programming
2008	CONFIG2	3FFF	-	-	-	-
		1	BOR4V	BOR40V	Brown-out Reset Selection bit	Brown-out Reset set to 4.0V
		3	WRT	OFF	Flash Program Memory Self Write Enable bits	Write protection off

Figure 10 Configuration Bits generated using MPLAB X

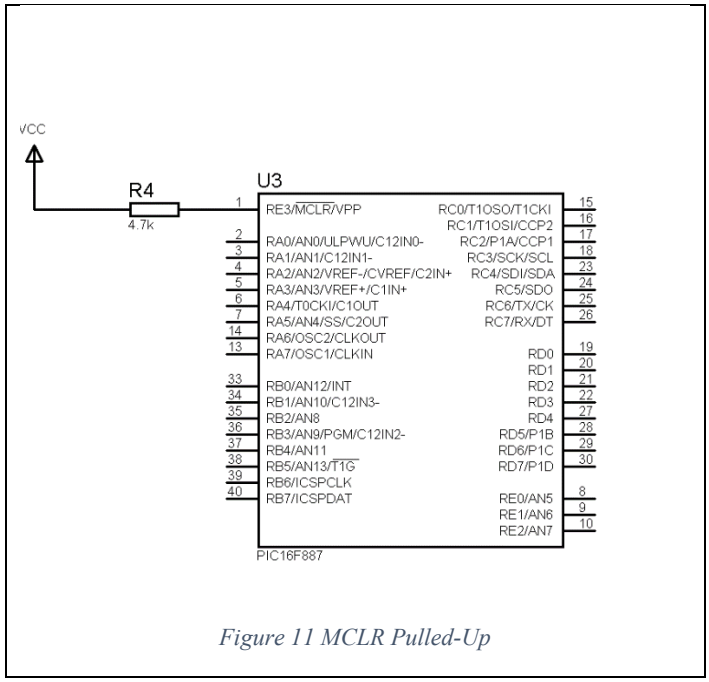


Figure 11 MCLR Pulled-Up

2.2 Display driver

2.2.1 Introduction to MAX7219 display driver

For interfacing various 7 segment displays in the project MAX7219 chip is used. It supports interfacing up to 8 displays and can control each segments individually. It is compatible with SPI protocol. Brightness of each display can be controlled and a test mode is provided to check if the displays are working properly or not. It has an inbuilt oscillator which supports 10MHz serial interface. MAX7219 is usually used to drive Common-Cathode LED displays.

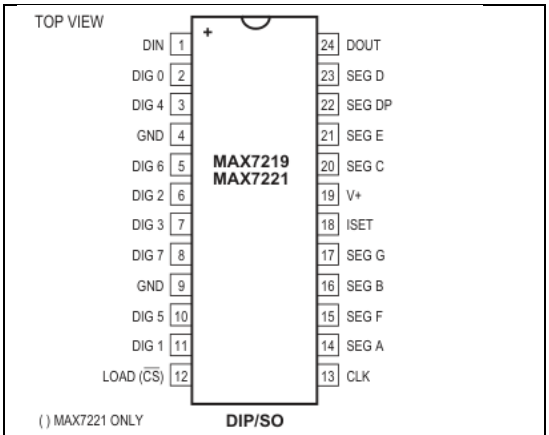


Figure 12 Pinout of MAX7219

Using 4 wire serial interface (SPI), displays can be controlled intelligently. MAX7219 works on 5V with maximum current up to 330mA. Display scan rate can go from 500Hz till 1300Hz.

PIN	NAME	FUNCTION
1	DIN	Serial-Data Input. Data is loaded into the internal 16-bit shift register on CLK's rising edge.
2, 3, 5–8, 10, 11	DIG 0–DIG 7	Eight-digit drive lines that sink current from the display common cathode. The MAX7219 pulls the digit outputs to V+ when turned off. The MAX7221's digit drivers are high-impedance when turned off.
4, 9	GND	Ground. Both GND pins must be connected.
12	LOAD (MAX7219)	Load-Data Input. The last 16 bits of serial data are latched on LOAD's rising edge.
	$\overline{CS}$ (MAX7221)	Chip-Select Input. Serial data is loaded into the shift register while $\overline{CS}$ is low. The last 16 bits of serial data are latched on $\overline{CS}$'s rising edge.
13	CLK	Serial-Clock Input. 10MHz maximum rate. On CLK's rising edge, data is shifted into the internal shift register. On CLK's falling edge, data is clocked out of DOUT. On the MAX7221, the CLK input is active only while $\overline{CS}$ is low.
14–17, 20–23	SEG A–SEG G, DP	Seven Segment Drives and Decimal Point Drive that source current to the display. On the MAX7219, when a segment driver is turned off it is pulled to GND. The MAX7221 segment drivers are high-impedance when turned off.
18	ISET	Connect to V_{DD} through a resistor (R_{SET}) to set the peak segment current (Refer to *Selecting R_{SET} Resistor and Using External Drivers* section).
19	V+	Positive Supply Voltage. Connect to +5V.
24	DOUT	Serial-Data Output. The data into DIN is valid at DOUT 16.5 clock cycles later. This pin is used to daisy-chain several MAX7219/MAX7221's and is never high-impedance.

Figure 13 MAX7219 Pin Description

Data is sent in 16-bit packet. Each bit is shifted (into DIN) at the rising edge of the CLK. Data is latched on the rising edge of the LOAD/CS pin. LOAD/CS must go high concurrently or after 16th rising CLK edge but before the 17th rising CLK edge.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

Figure 14 Serial-Data Format MAX7219

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xXF

Figure 15 MAX7219 Register Address Map

MAX7219 consists of 14 addressable digit and control registers. Digit register which ranges from 0x00 to 0x08 can be used to configure and update individual digit displays. Control Registers range from 0x09 till 0x0F. It consists of Decode Mode Register<0x09> which sets BCD code B or no-decode operation. If decode mode is enabled, the decoder checks the lower nibble of the data in the digit register (D3 – D0) and D7 is used for setting the Decimal Point. If no decode format is selected, data bits D7 – D0 corresponds to the segment lines.

DECODE MODE	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7–0	0	0	0	0	0	0	0	0	0x00
Code B decode for digit 0 No decode for digits 7–1	0	0	0	0	0	0	0	1	0x01
Code B decode for digits 3–0 No decode for digits 7–4	0	0	0	0	1	1	1	1	0x0F
Code B decode for digits 7–0	1	1	1	1	1	1	1	1	0xFF

Figure 16 Decode-Mode Register Example MAX7219

7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6–D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

**The decimal point is set by bit D7 = 1*

Figure 17 Code B Format MAX7219

Shutdown Register <0x0C> is used to set the driver in Shutdown more or Normal Operation Mode. Setting the D0 bit as 1 makes the driver work in Normal Operation. In Shutdown mode the MAX7219 oscillator is halted and all displays are pulled to V+ which blanks the displays. Maximum current consumption in shutdown mode is 150µA. It takes 250µS for MAX7219 to leave the shutdown mode and enter the normal operation mode. Intensity can be controlled by using the Intensity Control Register <0x0A>. Initially the intensity is set by RSET resistor connected between V+ and ISET and the minimum value should be 9.53kΩ which sets the segment current at 40mA. For further control of the brightness Intensity Control register is used. The MAX7219 has an internal pulse-width modulator which is controlled by lower nibble of intensity register. The intensity value ranges from 0x00 (Min brightness) – 0x0F (Max brightness). Another important control register is the Scan-Limit Register <0x0B>. Scan limit sets how many digits are displayed from 1 to 8. As the digits are displayed in multiplexed manner with typical scan rate of 800Hz with 8 digits, if fewer digits are displayed the scan rate is 8Fosc/N where N is the number of digits scanned. On Initial Power-Up all the control registers are Reset, the display is blanked, display scan is one digit and the driver is in shutdown mode.

To minimize power supply ripple, a 10µF electrolytic capacitor and a 0.1uF ceramic capacitor is to be connected between the V+ and GND as close to device as possible. For calculation of RSET and Power Dissipation calculation refer Page 33 **MAX7219 Electrical Characteristics**.

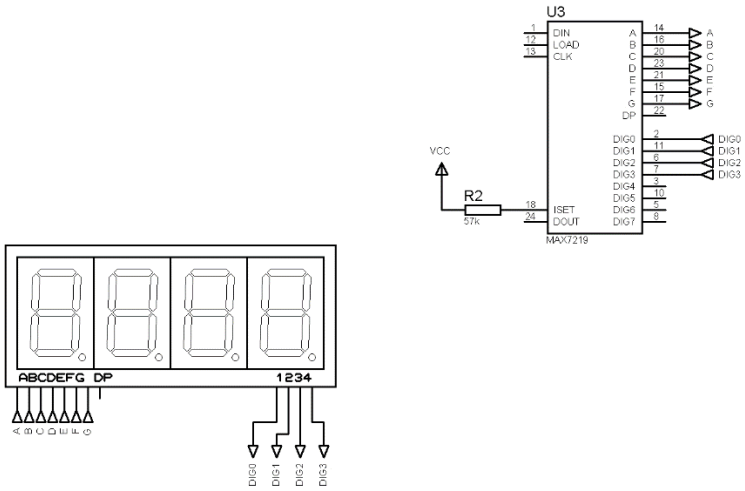


Figure 18 MAX7219 Interfacing with 7 Segment Display

2.2.2 Interfacing MAX7219 with PIC16F887 using SPI protocol

PIC16F887 consist of Master Synchronous Serial Port (MSSP) module which can be used for Serial communication with other devices. MSSP can be operated in SPI or I2C mode. For interfacing with MAX7219 display driver SPI protocol is used. For using MSSP SSPSTAT, SSPCON & SSPCON2 registers are used. SPI mode supports 8 bits of data to be synchronously transmitted and received. For communication with MAX7219 4 pins are used:

- Serial Data Out (SDO) – RC5
- Serial Data In (SDI) – RC4
- Serial Clock (SCK) – RC3
- Latching the Data (SS) – RD0

In order to configure MSSP in SPI mode, SSPCON and SSPSTAT bits have to be programmed. For interfacing with MAX7219, PIC16F887 is used in Master Mode. MSSP has a transmit/receive shift register SSPSR and a buffer register SSPBUF. SSPSR shifts the data out (MSB first). SSPSR is not directly readable or writable and can only be addressed through SSPBUF. In order to configure MSSP in SPI mode, SSPEN bit in SSPCON must be set. Default mode of MSSP on reset is SPI Master Mode with CLK = Fosc/4. The corresponding SDI pin is automatically controlled by MSSP while SDO pin has to be configured as an input by clearing the TRISC bit. SCK pin must have TRISC bit as set. For the project SS pin is not utilized so it is disabled by clearing the corresponding TRISC bit and is configured as LOAD pin for MAX7219.

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

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

bit 7 **SMP:** Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In I^2C Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit

CKP = 0:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK

CKP = 1:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit

(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit

(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/Write bit information (I^2C mode only)
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not ACK bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
Receive (SPI and I^2C modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit (I^2C mode only):
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is empty

Figure 19 SSPSTAT SSP Status Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

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

bit 7 **WCOL:** Write Collision Detect bit
Master mode:
1 = A write to the SSPBUF register was attempted while the I^2C conditions were not valid for a transmission to be started
0 = No collision
Slave mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit
In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register (must be cleared in software).
0 = No overflow
In I^2C mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output
In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In I^2C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I^2C Slave mode:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I^2C Master mode:
Unused in this mode

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = FOSC/4
0001 = SPI Master mode, clock = FOSC/16
0010 = SPI Master mode, clock = FOSC/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0110 = I^2C Slave mode, 7-bit address
0111 = I^2C Slave mode, 10-bit address
1000 = I^2C Master mode, clock = FOSC / (4 * (SSPADD+1))
1001 = Load Mask function
1010 = Reserved
1011 = I^2C firmware controlled Master mode (Slave idle)
1100 = Reserved
1101 = Reserved
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

Figure 20 SSPCON Register

In MAX7219, data is latched on the rising edge of the LOAD/CS pin. So PIC16F887 PORTA Pin 5 is connected to LOAD/CS pin in MAX7219 for latching the data once all the packet is transferred. In MAX7219, as the data is moved in during the rising edge of the CLK, in the SSPSTAT CKE bit is set to make the data transmission during the rising edge of CLK. CKP is default 0 which makes sure the default level of CLK is 0V. BF bit (Buffer Full) is polled in the SSPSTAT register of PIC16F887 to ensure the data transmission is complete before the new data is sent.

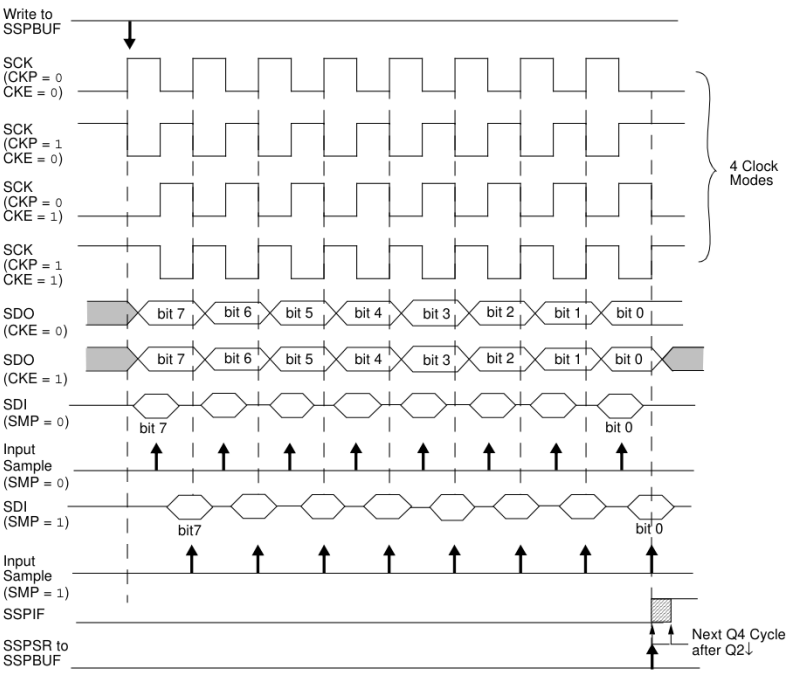


Figure 21 SPI Master Mode Waveform

MAX7219 and PIC16F887 SPI Function Snippet

Code Snippet 1 SPI data transfer function (to MAX7219)

```
/*
  SPI communication between PIC16F887 and MAX7219
  Call SPI_tx(addr,data) with Address of the register and the corresponding
        data to be sent.
*/

void SPI_tx(unsigned char addr,unsigned char dat)
{
    PORTA.F5 = 0;//LOAD_CS pin is made low
    SSPBUF = addr;//Send the address first
    while(~BF_bit);//Wait for transmission to complete
    SSPBUF = dat;//Send the data
    while(~BF_bit);//Wait for Transmission
    PORTA.F5 = 1;//Latch the data by pulling up LOAD_CS pin
}
```

MAX7219 display character ‘1’

Code Snippet 2 Display Character '1' in MAX7219

```
void main()
{
    TRISC.F5 = 0;  //SDO
    TRISC.F3 = 0;  //SCK
    TRISA.F5 = 0; //Configured as LOAD pin
    SSPCON = SSPCON | 0x22;
    SSPSTAT = SSPSTAT | 0xC0;
    tx_max(0x09, 0xFF);  //decoding
    tx_max(0x0A,0x0F); //intensity
```

```
    tx_max(0x0B, 0x03); //scan limit
    tx_max(0x01,0x01);
    tx_max(0x02,0x01);
    tx_max(0x03,0x01);
    tx_max(0x04,0x01);
    tx_max(0x0C, 0x01); //disable sleep
    while(1);
}
```

2.2.3 Brightness control using ADC in PIC16F887

PIC16F887 consists of a 10-bit ADC. It uses successive approximation and stores the result in ADRESL and ADRESH registers. ADC reference can be VDD or a voltage applied to a specific pin which can be defined in the software. PIC16F887 has 14 Analog compatible pins. At a time only one pin can be connected to the sample and hold circuit. CHS bits in ADCON0 is used to select the pin for reading the ADC value. To enable the ADC peripheral ADCON bit in ADCON0 is to be set. Once the peripheral is enabled a required amount of time has to elapsed (also called A/D acquisition time) before the conversion can be started. For A/D Acquisition time calculation refer Page 34 **A/D Acquisition Time**.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

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

bit 7-6 **ADCS<1:0>:** A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)

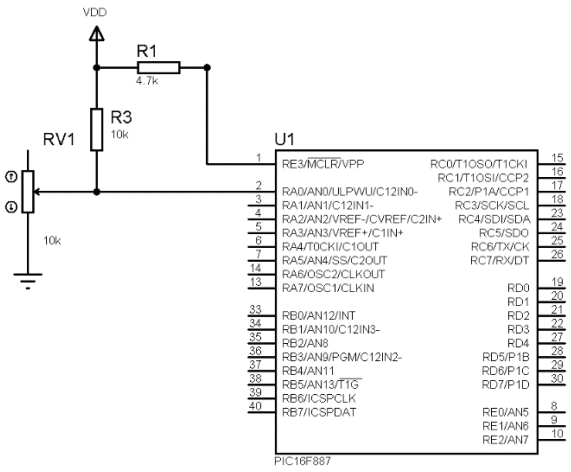
bit 5-2 **CHS<3:0>: Analog Channel Select bits**
0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = AN12
1101 = AN13
1110 = CVREF
1111 = Fixed Ref (0.6 volt fixed reference)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Figure 22 ADCON0 Register

GO/DONE bit in ADCON0 is polled to wait for the conversion to finish. POT is connected to RA0/AN0 pin. The default channel selected on reset is AN0. 10K POT is connected in voltage divider configuration with a 10kΩ resistor.



With Vin = 5V, POT of 10kΩ is used, the Vout range according to voltage divider equation is between 0V to 2.5V. 10 bit ADC value is stored in ADRESH:ADRESL with Left Justification (ADFM = 0 on Reset. Defined in ADCON1 register).

PIC16F887 will calculate the voltage at AN0 and accordingly send the brightness adjustment commands to MAX7219. As ADC is 10 bit, $2^{10} = 1024$. ADRESH:ADRESL can therefore range from 00000000:00xxxxxx to 11111111:11xxxxxx. MAX7219 defines intensity range from 0x00 (min) to 0x0F (max). It is desired that the brightness should be increased when the resistance of the POT increases. In order to achieve this the 16 brightness levels in MAX7219 are mapped to ADC levels in PIC16F887. As there are only 16 levels allowed for adjusting the brightness in MAX7219, only the ADRESH resistor values are used. This even if reduces the resolution of ADC to 8 bits, the brightness level that display driver can achieve makes 8 bit a viable option reducing the programming complexity which will come with using variable above 8-bit size. The mapping is done by the equation *ADRESH * 15 / 128*. The corresponding brightness level is then adjusted by writing in the intensity control register 0x0A in MAX7219.

2.2.4 Simulation Results

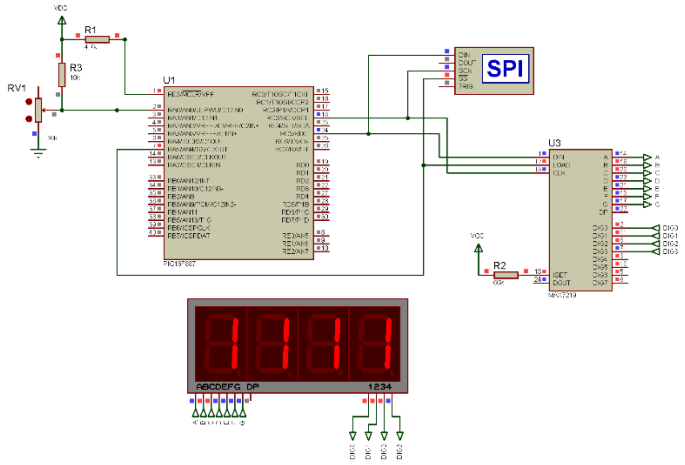
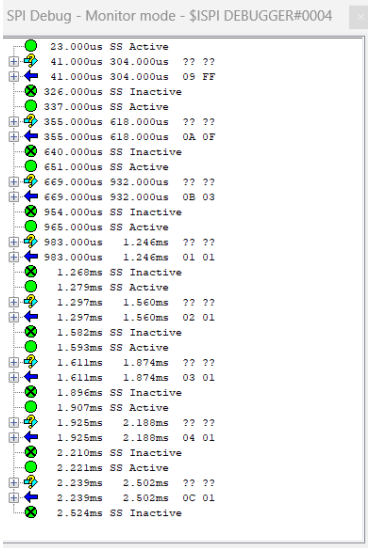


Figure 23 Interfacing PIC16F887 with MAX7219 and POT



SPI debugger utility in Proteus shows different commands that were sent to the MAX7219 IC over SPI lines. All the four displays were initialized to character '1'.

2.3 Real Time Clock

2.3.1 Introduction to RTC DS1307

As stated in the datasheet the DS1307 is a serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2C. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information though for this project only the minutes and hours are used. RTC can be operated in 24-hour or 12-hour mode. It also has a backup battery so in the event of power failure it switches automatically to backup battery power. DS1307 also has a 57-byte battery backed-up RAM.

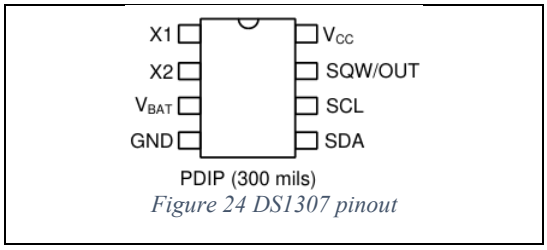


Figure 24 DS1307 pinout

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1. **Note:** For more information on crystal selection and crystal layout considerations, refer to *Application Note 58: Crystal Considerations with Dallas Real-Time Clocks*.
2	X2	
3	V_{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V_{BAT} pin may prevent proper operation. If a backup supply is not required, V_{BAT} must be grounded. The nominal power-fail trip point (V_{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/qa/info/ul/.
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I^2C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}.
6	SCL	Serial Clock Input. SCL is the clock input for the I^2C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}.
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V_{CC} or V_{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}. If not used, this pin can be left floating.
8	V_{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V_{CC} is below V_{TP}, read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

Figure 25 DS1307 pin description

A typical 3V battery is used with the DS1307. When VCC falls below 1.25 x VBAT, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When VCC falls below VBAT, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to VCC when VCC is greater than VBAT +0.2V and recognizes inputs when VCC is greater than 1.25 x VBAT.

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

Figure 26 DS1307 Registers

The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space. The contents of the time and calendar registers are in the BCD format. The CH bit in register 00h is default to 1 which states that the DS1307 oscillator is turned off. To enable the time keeping function CH bit should be cleared.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

Bits 1 and 0: Rate Select (RS[1:0]). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

Figure 27 DS1307 Control Register

For other electrical specifications and crystal consideration for DS1307 refer Section 3.4 in page 35.

2.3.2 Interfacing with PIC16F887 and I2C protocol

PIC16F887 and RTC DS1307 communicates with each other using I2C protocol. The MSSP module in PIC16F887 is used as I2C master mode with 7-bit addressing as DS1307 supports 7 bit addressing only. The 7-bit address of DS1307 as given in the datasheet is 1101000. In PIC16F887 two pins are used for data transfer RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the

data (SDA). The general idea is that the DS1307 RTC is configured with Square Wave Output of 1 Hz. The SQWOut pin of DS1307 is then connected to the T0CKI pin of PIC16F887. The timer module of PIC is enabled in the counter mode. The counter counts for 60 times then an interrupt is generated. This interrupt is serviced by reading the DS1307 time registers followed by updating the display driven by MAX7219 driver. The calculation of I2C pullup resistor is given in Section 3.4.2 Page 37. For ease in designing the project, DS1307 HW111 board is used which has a DS1307 and AT24CP module with pull-ups of 3.3k (for SDA, SCL and SQOut) and 32.768kHz crystal.

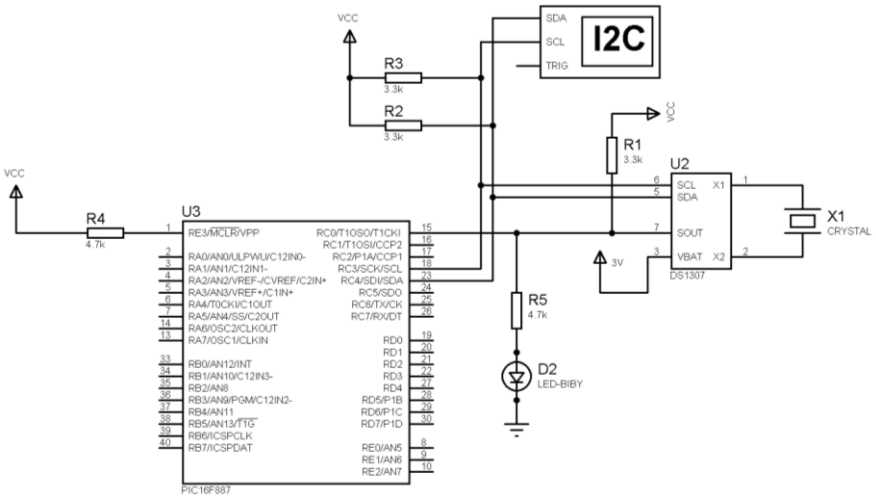


Figure 28 PIC16F887 and DS1307 Interfacing

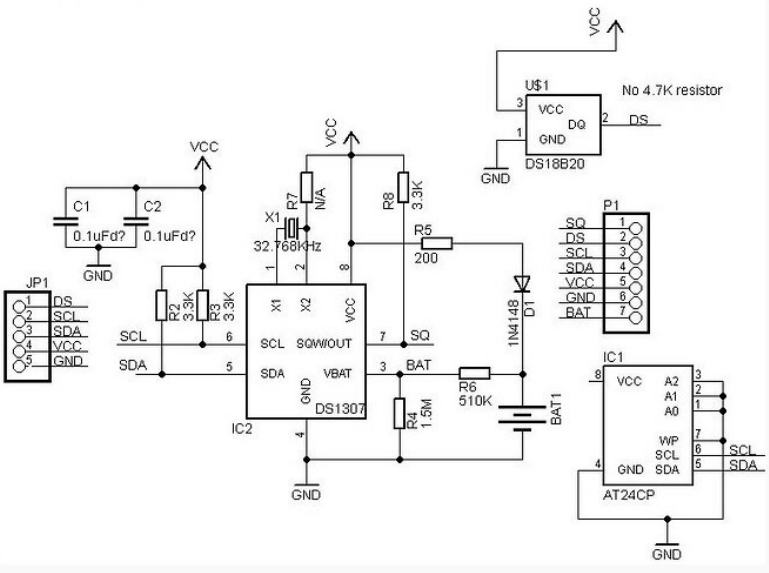


Figure 29 HW111 breakout board schematic (For DS1307)

PIC16F887 MSSP I2C master module is supported by interrupt generation. The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

- Acknowledge transmit
- Repeated Start condition

Master mode is enabled by setting the SSPM bits as 1000 (Master mode) in SSPCON register given in Figure 20 SSPCON Register page 13.

The max SCL line clock speed supported by DS1307 as given in the electrical specification is 100kHz. The baud rate generator in PIC16F887 for master mode is given as Fosc/(4 * (SSPADD + 1)) where SSPADD bits <6:0> are used for auto reload in BRG of PIC16F887. The recommended value of SSPADD considering the Fosc of PIC16F887 at 4MHz is 0x15 which is 21 in decimal. The corresponding SCL line speed will be roughly 45kHz which is fast enough for DS1307 to interpret the SCL and SDA line.

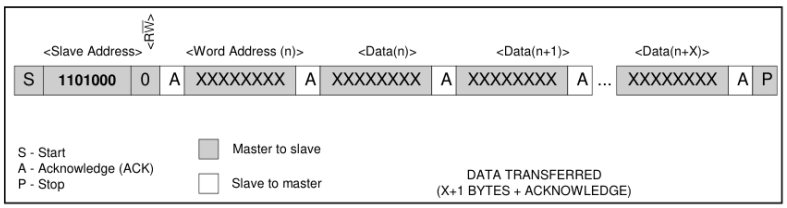
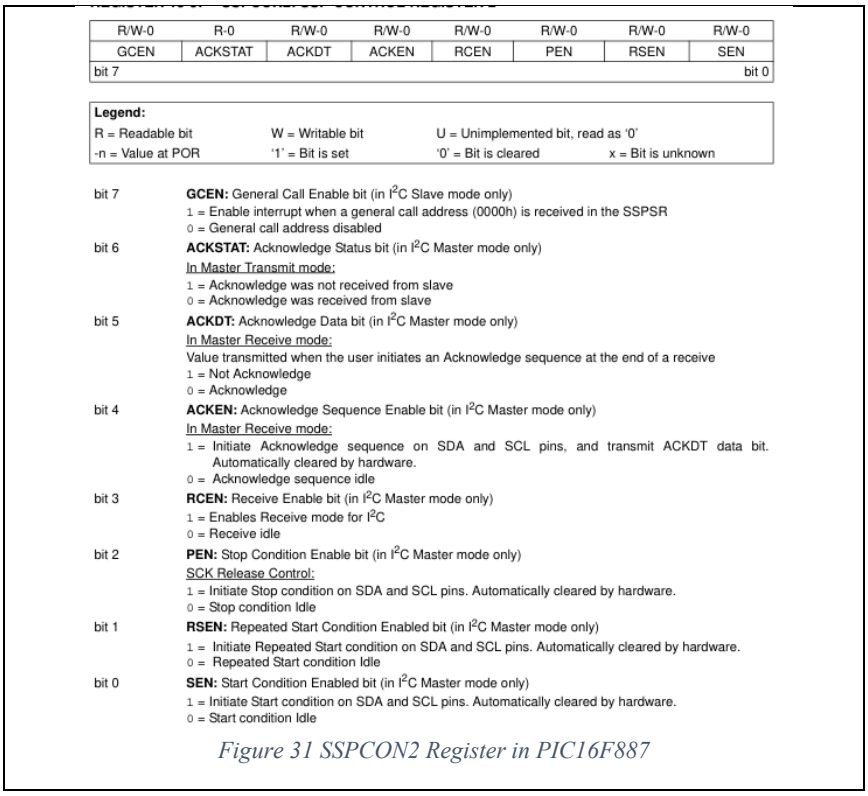


Figure 30 Writing to DS1307 registers

Different I2C conditions can be automated in the hardware using the SSPCON2 register.



R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

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

bit 7 **GCEN:** General Call Enable bit (in I²C Slave mode only)
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in I²C Master mode only)
In Master Transmit mode:
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT:** Acknowledge Data bit (in I²C Master mode only)
In Master Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in I²C Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in I²C Master mode only)
1 = Enables Receive mode for I²C
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in I²C Master mode only)
SCK Release Control:
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enabled bit (in I²C Master mode only)
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enabled bit (in I²C Master mode only)
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition Idle

Figure 31 SSPCON2 Register in PIC16F887

1Hz SQWOut enable code snippet

Code Snippet 3 Enable 1Hz SQWOutput on DS1307

```
SSPCON = SSPCON | 0x28;   //Configure in I2C master mode
SSPADD = 0x15;            //Clock speed of 90kHz
```

```
SEN_bit = 1;    //Enable a start condition
while(SEN_bit); //wait for start condition to complete

SSPBUF = 0xD0;  //RTC add with write command
while(SSPSTAT & 0x04); //Wait till transfer is complete

SSPBUF = 0x07;  //RTC SQWOut register address
while(SSPSTAT & 0x04); //Wait till transfer is complete

SSPBUF = 0x90;  //Enable 1Hz clock
while(SSPSTAT & 0x04); //Wait till transfer is complete

PEN_bit = 1;    //Enable a stop condition
while(PEN_bit); //wait for stop condition to complete
```

Timer1 module in PIC16F887 is configured as a 16-bit Counter with external clock source. The reason for using the 16-bit Timer1 module rather than a 8-bit Timer0 module is that the Timer1/Counter1 can still work with external clock source when the device is put to sleep. This decreases the current consumption as the microcontroller is off for about 60 seconds after which the overflow interrupt will wake up the microcontroller and the predetermined tasks can then be executed before the sleep function is executed again.

Timer1/Counter1 consists of register TMR1H:TMR1L pairs. External clock source is selected by making the TMR1CS bit as 1.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV(1)	TMR1GE(2)	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

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

bit 7 **T1GINV:** Timer1 Gate Invert bit(1)
1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 6 **TMR1GE:** Timer1 Gate Enable bit(2)
If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 is on if Timer1 gate is not active
0 = Timer1 is on

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale Value
10 = 1:4 Prescale Value
01 = 1:2 Prescale Value
00 = 1:1 Prescale Value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
1 = LP oscillator is enabled for Timer1 clock
0 = LP oscillator is off

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit
TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from T1CKI pin (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

Figure 32 T1CON Register in PIC16F887

Timer1 with external clock source acts as a Counter which increments during the rising edge of the external clock. For being able to use the Counter1 during the sleep, the T1SYNC bit in T1CON is to be set to 1. When the Timer1/Counter1 overflows from FFFFh to 0000h, TMR1IF flag in PIR1 register is set. TMR1H:TMR1L is configured as 0xFFC4 which will count the external clock rising edge 60 times and then produce an interrupt which will wake up the microcontroller from sleep and further tasks are executed. The GIE bit is not enabled as the program will continue executing the next instructions stated after the sleep instruction.

The *asm sleep;* command in mikroC is used to put the microcontroller into sleep mode. The WDT is disabled in config fuses so the microcontroller only wakes up after the 60th clock pulse on the T1CKI pin. The timer is turned on by setting the TMR1ON bit inT1CON and while updating the TMR1H:TMR1L registers the timer is turned off to prevent write contention error.

60 Seconds Counter with Sleep Mode

Code Snippet 4 60sec Counter with Sleep Mode

```
PORTC.F0 = 0; //Clear the T1CKI port pin
TRISC.F0 = 1; //Configure T1CKI as an Input pin

TMR1CS_bit = 1;//Use external clock. Configured as a Counter
T1SYNC_bit = 1;//Do not synchronize the external clock

TMR1IE_bit = 1;//Enable overflow interrupt
PEIE_bit = 1;//Enable Peripheral Interrupt

while(1) {
      //Load TMR1H:TMR1L with 0xFFC4 which makes the Counter overflow
      //     at 60th rising edge of external clock
      TMR1H = 0xFF;
      TMR1L = 0xC4;

      TMR1IF_bit = 0;//Clear Timer1/Counter1 overflow interrupt
      TMR1ON_bit = 1;//Turn on the Timer

      asm sleep;//put the microcontroller in sleep mode

      TMR1ON_bit = 0; //To prevent write contention turn off the timer
}
```

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE(1,3)	T0IF(2)	INTF	RBIF
bit 7							bit 0

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

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** INT External Interrupt Enable bit
1 = Enables the INT external interrupt
0 = Disables the INT external interrupt

bit 3 **RBIE:** PORTB Change Interrupt Enable bit(1,3)
1 = Enables the PORTB change interrupt
0 = Disables the PORTB change interrupt

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit(2)
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** INT External Interrupt Flag bit
1 = The INT external interrupt occurred (must be cleared in software)
0 = The INT external interrupt did not occur

bit 0 **RBIF:** PORTB Change Interrupt Flag bit
1 = When at least one of the PORTB general purpose I/O pins changed state (must be cleared in software)
0 = None of the PORTB general purpose I/O pins have changed state

Figure 33 INTCON Register PIC16F887

U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as '0'

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = A/D conversion complete (must be cleared in software)
0 = A/D conversion has not completed or has not been started

bit 5 **RCIF:** EUSART Receive Interrupt Flag bit
1 = The EUSART receive buffer is full (cleared by reading RCREG)
0 = The EUSART receive buffer is not full

bit 4 **TXIF:** EUSART Transmit Interrupt Flag bit
1 = The EUSART transmit buffer is empty (cleared by writing to TXREG)
0 = The EUSART transmit buffer is full

bit 3 **SSPIF:** Master Synchronous Serial Port (MSSP) Interrupt Flag bit
1 = The MSSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
SPI
A transmission/reception has taken place
I^2C Slave/Master
A transmission/reception has taken place
I^2C Master
The initiated Start condition was completed by the MSSP module
The initiated Stop condition was completed by the MSSP module
The initiated restart condition was completed by the MSSP module
The initiated Acknowledge condition was completed by the MSSP module
A Start condition occurred while the MSSP module was idle (Multi-master system)
A Stop condition occurred while the MSSP module was idle (Multi-master system)
0 = No MSSP interrupt condition has occurred

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode

bit 1 **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = A Timer2 to PR2 match occurred (must be cleared in software)
0 = No Timer2 to PR2 match occurred

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = The TMR1 register overflowed (must be cleared in software)
0 = The TMR1 register did not overflow

Figure 34 PIR1 Register PIC16F887

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as '0'

bit 6 **ADIE:** A/D Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

bit 3 **SSPIE:** Master Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** Timer2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

Figure 35 PIE1 Register PIC16F887

Although this much may be enough for updating the display driver after every 60th clock edge, the timer1 overflow value needs to be in sync with the DS1307 seconds register value. This ensures that the display is updated only during the 60th second of actual time interval in RTC. Therefore the initialization of the RTC makes sure that SQWOut is enabled with 1Hz frequency, the Timer1 is configured appropriately and the TMR1H:TMR1L register values are updated according to the seconds

information received from the RTC. As the content of RTC DS1307 is in BDC format the values are taken and converted to appropriate decimal format which is then used to update the TMR1H:TMR1L registers initially. In the preceding overflow the TMR1H:TMR1L is updated with 0xFFC4 value as usually because the microcontroller's overflow clock is in sync with the RTC seconds' register.

BCD to Decimal Conversion for TMR1H:TMR1L

Code Snippet 5 BCD to Decimal Conversion for TMR1H:TMR1L

```
unsigned char bcd = 0x43;              //BCD value to be converted (it will be received from RTC)
unsigned char val;                     //New value will be stored in val
unsigned char lower_nib = bcd & 0x0F;          //Get the lower nibble value, mask the upper nibble
val = (bcd >> 4)*10;                           //Left Shift the upper nibble and multiply by 10
val += lower_nib;                      //Resultant val = 43
val = 0xFF - (60 - val);                       //TMR1 overflow val = 238
```

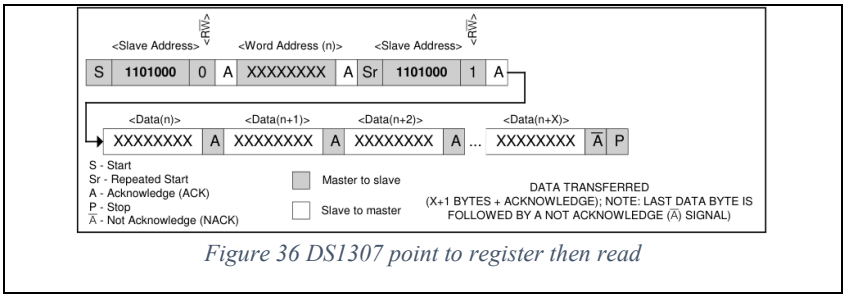
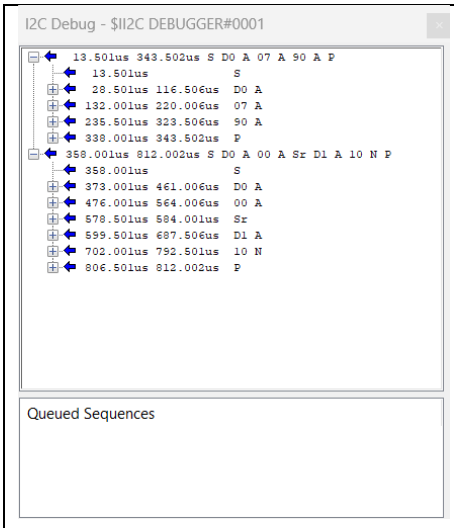


Figure 36 DS1307 point to register then read

According to the Figure 36 DS1307 point to register then read), the seconds register as given in the DS1307 Figure 26 DS1307 Registers) is at the location 0x00. PIC16F887 initially writes to the RTC with the specified address then a repeated start condition is passed by the microcontroller which is followed by read address to the RTC DS1307 slave. Here only the seconds register is read and the TMR1H:TMR1L registers of PIC16F887 are updated accordingly.

2.3.3 Simulation Results

Using the I2C debugger tool in Proteus, the I2C bus commands can be seen



The I2C debugger tool shows the different master conditions defined on the bus by PIC16F887. Initially the microcontroller gives a Start Condition (S) followed by the 7-bit slave address and 8th bit as 0 for write operation on the slave (Address becomes 0xD0). The master then sends the register address 0x07 which corresponds to the SQWOut Register of the DS1307 and it is configured in 1Hz as specified in the Figure 27 DS1307 Control Register. Then the Stop Condition (P) is given by the master. With each data transfer the slave responds with an Ack (A). After configuring the SQWOut, the seconds register is read by first setting the pointer to Seconds register (Address is 0x00) followed by repeated start condition (Sr) with Slave Address and Read Bit (0xD1). The corresponding seconds value in BCD is sent back. Then the Stop Condition is issued with Master giving a NACK.

2.4 Internet Time Synchronization

2.4.1 Introduction to ESP8266-01

The ESP8266 is a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of a Tensilica L106 32-bit micro controller unit (MCU) and a Wi-Fi transceiver. It is the Wi-Fi chip used to get the internet time. Synchronization between PIC16F887 and ESP-01 is achieved over the UART protocol. ESP-01 is programmed using Arduino IDE and CP2102 TTL UART Serial Converter.

2.4.2 Getting NTP time from the internet

The NTP is a protocol which is used to synchronize the clocks to internet time. It is a part of TCP/IP protocol. NTPClient.h header defines various functions that can be called to get the current time from the internet. Initially NTPClient object is created stating the offset of +5:30 hours (Indian IST). PIC16F887 expects the time to be in the BCD format as the registers of the DS1307 work in BCD. So the numbers are converted to BCD using the formula ((val / 10) << 4) | (val % 10) where val is the value to be converted to BCD.

ESP-01 NTPClient Initialization

Code Snippet 6 ESP-01 NTPClient Init

```
const long utcOffsetInSeconds = 19800; // India +5:30 .. (5hr * 60min + 30min) * 60sec
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);
timeClient.getHours() //Get the Hours
timeClient.getMinutes()//Get the Minutes
timeClient.getSeconds() //Get the Seconds
```

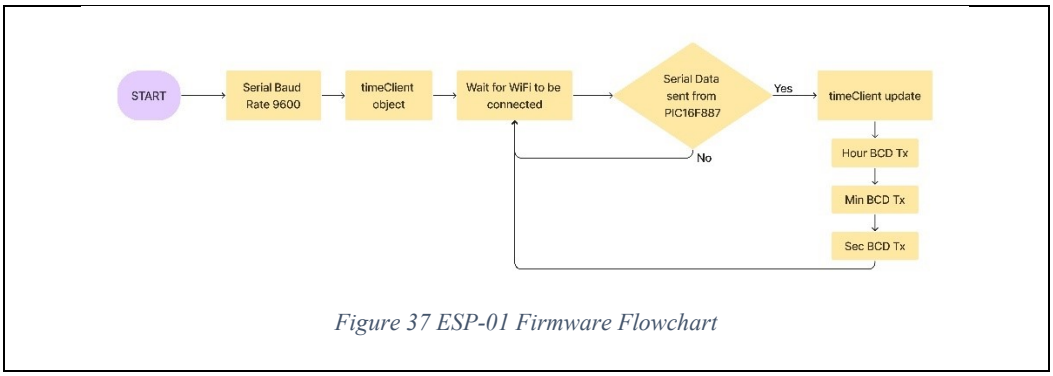


Figure 37 ESP-01 Firmware Flowchart

2.4.3 Interfacing PIC16F887 and ESP8266-01 using UART

ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (ESUART) Peripheral is used in Asynchronous Mode for achieving UART with ESP-01. To enable UART peripheral SPEN bit in RCSTA register should be set to 1. For achieving Async operations SYNC (In TXSTA) should be equal to 0. By wiring the characters to TXREG register, the data is sent (It required TXEN to be set). TRMT_bit is polled to make sure the data transmission is complete. Similarly Reception is enabled by setting the CREN bit. The baud rate for transmission is set to 9600 by setting the SPBRG register to 25 and enabling the high speed transfer mode by setting the BRGH_bit in TXSTA register. As ESP8266-01 works on 3.3V while PIC16F887 works on 5V, a Logic level converter based on BSS138 N-Channel MOSFET is used between the UART lines (Tx and Rx). The schematic of the Level Converter is given below

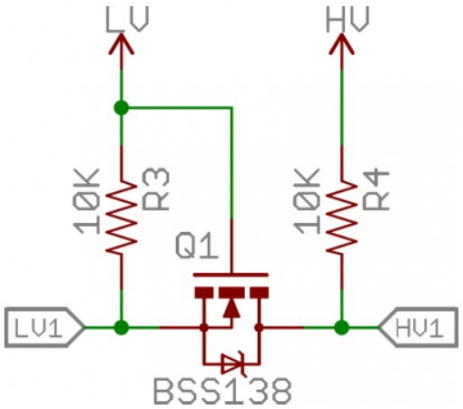


Figure 38 Bi-Level Converter for UART

BAUD RATE	Fosc = 4.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—
1200	1202	0.16	207
2400	2404	0.16	103
9600	9615	0.16	25
10417	10417	0.00	23
19.2k	19.23k	0.16	12
57.6k	—	—	—
115.2k	—	—	—

Table shows the baud rate that can be achieved with Internal Oscillator set to 4MHz. The Error Rate is at 0.16% which is considerably less for any error to take place.

RCREG register is used to read the contents that are sent from the ESP-01.

UART Configure in PIC16F887

Code Snippet 7 UART Configure in PIC16F887

```
TRISC = TRISC | 0xC0;
    SPBRG = 25; //for 4MHz OSC 9600 baud with 0.16% error
    BRGH_bit = 1;//Enable high speed mode
    SPEN_bit = 1;    //enable uart
    TXEN_bit = 1; //enable transmission
    CREN_bit = 1; //enable reception
    while(1) {
          TXREG = 1;
          while(!TRMT_bit); //wait for transmission to complete
          while(!RCIF_bit); //wait for reception
           //Read RCREG register and get the time data
     }
```

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

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

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
Synchronous mode:
Don't care

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Figure 39 TXSTA Register in PIC16F887

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

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

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

Figure 40 RCSTA Register in PIC16F887

ESP-01 requires some basic connections to be made before it can be used. The Serial data is read inside the made loop waiting for PIC16F887 to send some data. When the PIC is in Time Sync Mode it send a command with value 0x01 to ESP-01. ESP then updates its clock with the internet time using NTPClient object and sends BCD format of the time to PIC16F887. First the Hour followed by Minutes followed by Seconds are sent to PIC microcontroller. PIC reads all the time information and then updates the DS1307 registers. After that MAX7219 driver is updated according to the latest time and counter starts according to the seconds received. PIC then goes back to Sleep mode.

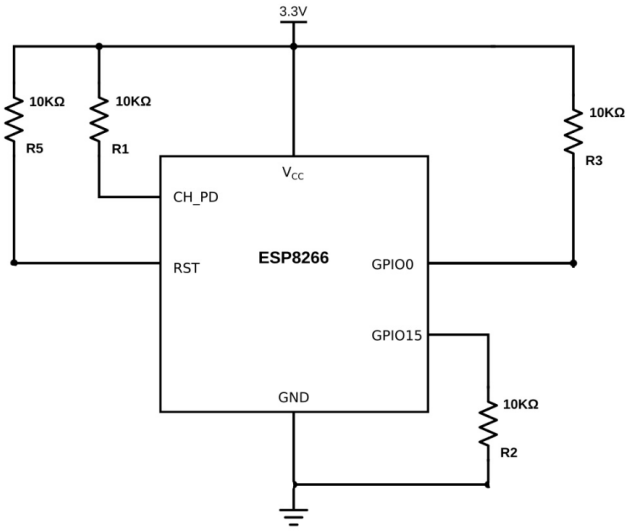


Figure 41 ESP-01 Basic Connection Diagram

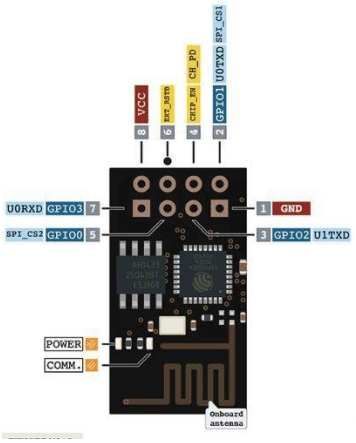


Figure 42 ESP-01 Pinout

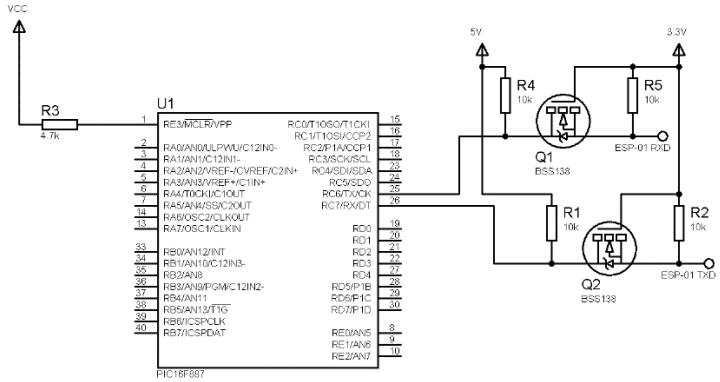


Figure 43 PIC16F887 UART Connection

Here TX and RX from PIC16F887 goes to RXD and TXD in ESP-01 respectively.

2.5 Using Switch for setting time manually and all peripheral integration

Two Switches are used to achieve manual time setting mode, synchronization with the internet time, going into brightness adjustment mode. These are all referred to as task being performed by PIC16F887. When no tasks are left to perform the microcontroller goes into sleep mode. A white LED is connected to pin RA1 which turns on when the microcontroller performs some task and turns off when the controller is in sleep mode. Majority of the time it will be in the sleep mode and keeps counting the SQWOut pin from DS1307 using async counter. After 60 counts the controller comes out of the sleep mode and reads the DS1307 time registers followed by updating the MAX72019 display drivers. Then the controller goes back into sleep mode. Switches can also be used to wake up the microcontroller so that it will perform the specific task. Two switches are given for this purpose. SW0 and SW1. SW0 is used to change the state in which microcontroller is (wake up from sleep or go back to sleep). The condition of the SW1 switch determines which series of tasks will be performed by the microcontroller.

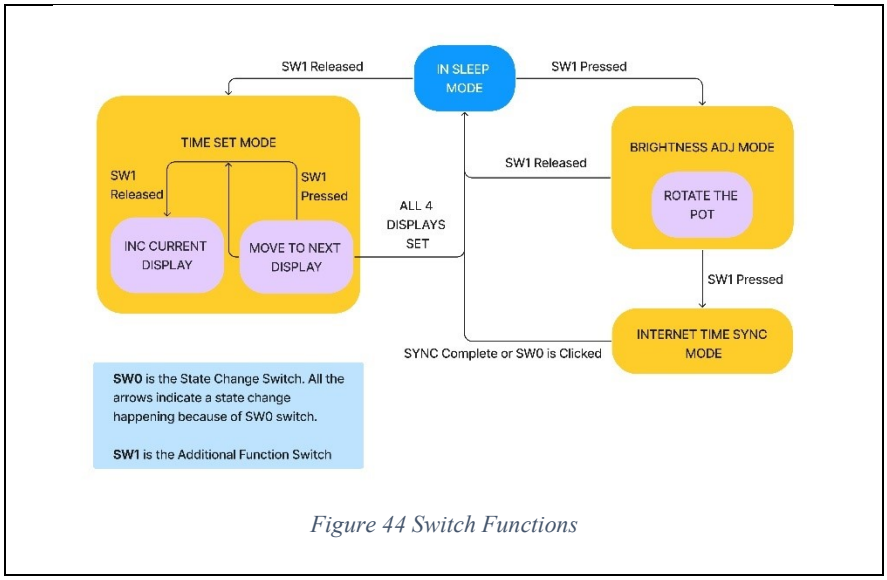


Figure 44 Switch Functions

SW0 and SW1 are connected to RB0/INT and RB1. SW0 therefore makes the state change which SW1 decides the functionality. Both the port pin are configures as digital input and have weak pull ups enables from OPTION register. Both are connected by 0.1uF ceramic capacitor individually to remove debounce effect. INTE_bit is used to enable the external interrupt and INTEDG_bit is cleared to make the interrupt on falling edge of the click.

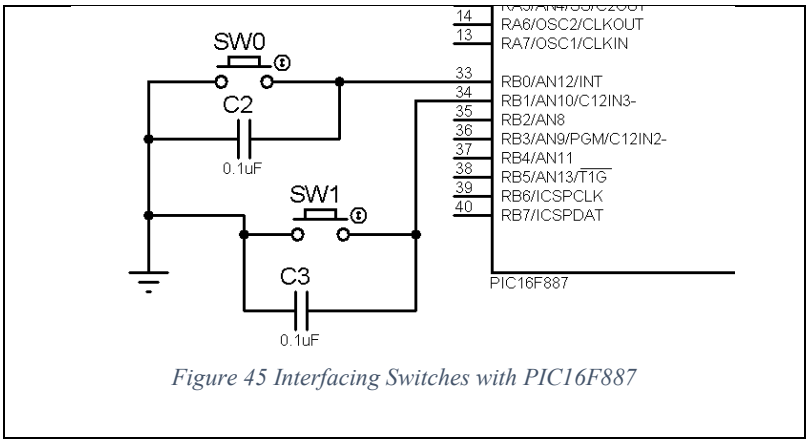


Figure 45 Interfacing Switches with PIC16F887

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

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

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Figure 46 OPTION Register in PIC16F887

Switch Functions Code Snippet

Code Snippet 8 Switch Function

```
while(1) {
        PORTA.F1 = ~PORTA.F1;//Toggle MODE LED
        asm sleep;
        delay_ms(20);//Debounce
        INTF_bit = 0;//Clear the INT flag
        PORTA.F1 = ~PORTA.F1;//Toggle MODE LED
        //Check if 60 seconds counter interrupted
        if (TMR1IF_bit) {
                //60 counts completed. Read DS1307 registers and update MAX7219
         }
         //Check if SYNC/SET switch was clicked while INC switch is clicked
         else if (PORTB.F1) {
                //In time set mode
         }
         else {
           //Brightness adjust mode
                        //ADC commands while loop
                //loop break because of SW0 click
                if (~PORTB.F1) {
                        //INC button was clicked. In NTP SYNC mode
                }
                Else {
                        //Exit and go back to sleep mode}
        }
        Configure normal operations
        Return to normal mode
}
```

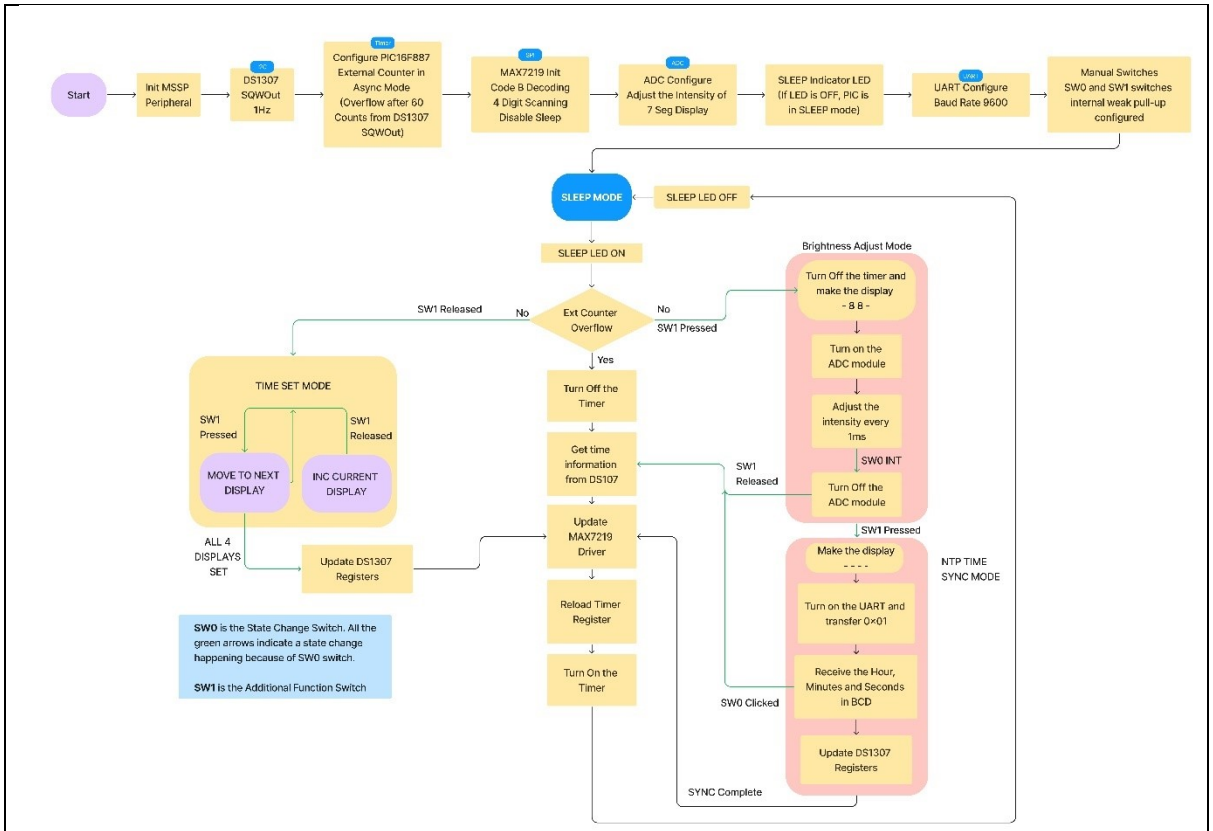


Figure 47 PIC Firmware Flowchart

3 Components and Electrical Specifications

3.1 Bill of Materials

Component	Value	Quantity	Total Cost (Rs)
PIC16F887 ~I/P	DIP-40	1	299
HW111 DS1307	-	1	49
MAX7219CNG	DIP-28	1	90
7 Segment Display	0.56 Inch 0.8 Inch	2 Each	30
IC Holders	24 pin Wide 28 pin Narrow 40 pin	1 5 1	30
ESP-01	-	1	150
BSS138 Bi-Directional level converter	-	1	41
AMS1117		1	35
Resistor	4.7k 10k 57k	3 4 1	20
Capacitor	10uF Electrolytic 0.1uF Ceramic	2 1	8
General Purpose PCB	6 * 4 Inch		40
Rotary POT	10k		10
Square Switch	-	2	15
Single Strand	2m	1	15
PCB Connector	2 Terminal	1	10
JST Connector	4 Wire	2	20

3.2 MAX7219 and 7 Segment Display

3.2.1 7 Segment Display Absolute Ratings

Absolute Maximum Rating at Ta=25℃

Parameter	AlGaAs Red	Bright Red	Green	Yellow	Red Orange	Unit
Power Dissipation Per Segment	75	40	75	60	75	mW
Peak Forward Current Per Segment (1/10 Duty Cycle, 0.1ms Pulse Width)	125	60	100	80	100	mA
Continuous Forward Current Per Segment Derating Linear from 25℃ Per Segment	30 0.4	15 0.2	25 0.33	20 0.27	25 0.33	mA mA/℃
Reverse Voltage Per Segment	5	5	5	5	5	V
Operating Temperature Range	-35℃ to +85℃					
Stroage Temperature Range	-35℃ to +85℃					

Figure 48 7 Segment Display Absolute Maximum Ratings

3.2.2 MAX7219 Electrical Characteristics, RSET and Power Dissipation Calculation

Voltage (with respect to GND)
V+ .....-0.3V to 6V
DIN, CLK, LOAD, $\overline{CS}$ .....-0.3V to 6V
All Other Pins.....-0.3V to (V+ + 0.3V)
Current
DIG 0–DIG 7 Sink Current.....500mA
SEG A–G, DP Source Current 100mA
Continuous Power Dissipation (T_A = +85°C)
Narrow Plastic DIP (derate 13.3mW/°C above +70°C) 1066mW
Wide SO (derate 11.8mW/°C above +70°C).....941mW
Narrow CERDIP (derate 12.5mW/°C above +70°C)... 1000mW

Operating Temperature Ranges (T_{MIN} to T_{MAX})
MAX7219C_G/MAX7221C_G .....0°C to +70°C
MAX7219E_G/MAX7221E_G..... -40°C to +85°C
Storage Temperature Range .....-65°C to +160°C
Lead Temperature (soldering, 10s) .....+300°C

($V+$ = 5V ±10%, R_{SET} = 9.53kΩ ±1%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Operating Supply Voltage	V+		4.0		5.5	V
Shutdown Supply Current	I+	All digital inputs at V+ or GND, T_A = +25°C			150	µA
Operating Supply Current	I+	R_{SET} = open circuit			8	mA
		All segments and decimal point on, $I_{SEG\_}$ = -40mA		330		
Display Scan Rate	f_{OSC}	8 digits scanned	500	800	1300	Hz
Digit Drive Sink Current	I_{DIGIT}	V+ = 5V, V_{OUT} = 0.65V	320			mA
Segment Drive Source Current	I_{SEG}	T_A = +25°C, V+ = 5V, V_{OUT} = (V+ - 1V)	-30	-40	-45	mA
Segment Current Slew Rate (MAX7221 only)	$\Delta I_{SEG}/\Delta t$	T_A = +25°C, V+ = 5V, V_{OUT} = (V+ - 1V)	10	20	50	mA/µs
Segment Drive Current Matching	ΔI_{SEG}			3.0		%
Digit Drive Leakage (MAX7221 only)	I_{DIGIT}	Digit off, V_{DIGIT} = V+			-10	µA
Segment Drive Leakage (MAX7221 only)	I_{SEG}	Segment off, V_{SEG} = 0V			1	µA
Digit Drive Source Current (MAX7219 only)	I_{DIGIT}	Digit off, V_{DIGIT} = (V+ - 0.3V)	-2			mA
Segment Drive Sink Current (MAX7219 only)	I_{SEG}	Segment off, V_{SEG} = 0.3V	5			mA

Figure 49 MAX7219 Electrical Specification

Current through each segment is 100 times more than the current in ISET. MAX7219 maximum segment current is 40mA.

I_{SEG} (mA)	V_{LED} (V)				
	1.5	2.0	2.5	3.0	3.5
40	12.2	11.8	11.0	10.6	9.69
30	17.8	17.1	15.8	15.0	14.0
20	29.8	28.0	25.9	24.5	22.6
10	66.7	63.7	59.3	55.4	51.2

Note: *R_{SET} values are in Kilo Ohms (kΩ)*

Figure 50 RSET vs Segment Current and LED forward Voltage

Consider 4 digits of HDSP-H103 is used. With maximum segment current at 15mA and forward voltage of 1.8V, appropriate RSET value according to the table is from 50kΩ – 66.7kΩ. Supply voltage is 5V so resistor capable of handling 0.032W power is required.

Power Dissipation of MAX7219 can be calculated using the following equation

PD = (V+ x 8mA) + (V+ - VLED)(DUTY x ISEG x N)

where:

V+ = supply voltage

DUTY = duty cycle set by intensity register

N = number of segments driven (worst case is 8)

VLED = LED forward voltage

ISEG = segment current set by RSET

Consider V+ as 5V, DUTY cycle is set to maximum (31/32), number of digits are 4, VLED is 1.8V and [i]segment current is 10mA.

Total Power Dissipation will be PD = (5 x 8mA)+(5-1.8)(31/32 x 10mA x 8) = 0.248W.

3.3 A/D Acquisition Time

As given in the PIC16F887 datasheet, Acquisition time is given as

Assumptions: $Temperature = 50°C$ *and external impedance of* $10k\Omega$ *5.0V* V_{DD}

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\mu s + T_C + [(Temperature - 25°C)(0.05\mu s/°C)]$$

The value for T_C *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{2047}\right) = V_{CHOLD} \quad ;[1]\ V_{CHOLD}\ \text{charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD}\ \text{charge response to}\ V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{2047}\right) \quad ;\text{combining [1] and [2]}$$

Solving for T_C*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/2047)$$
$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.0004885)$$
$$= 1.37\mu s$$

Therefore:

$$T_{ACQ} = 2\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.67\mu s$$

Here it is considered that the maximum source impedance is 10kΩ.

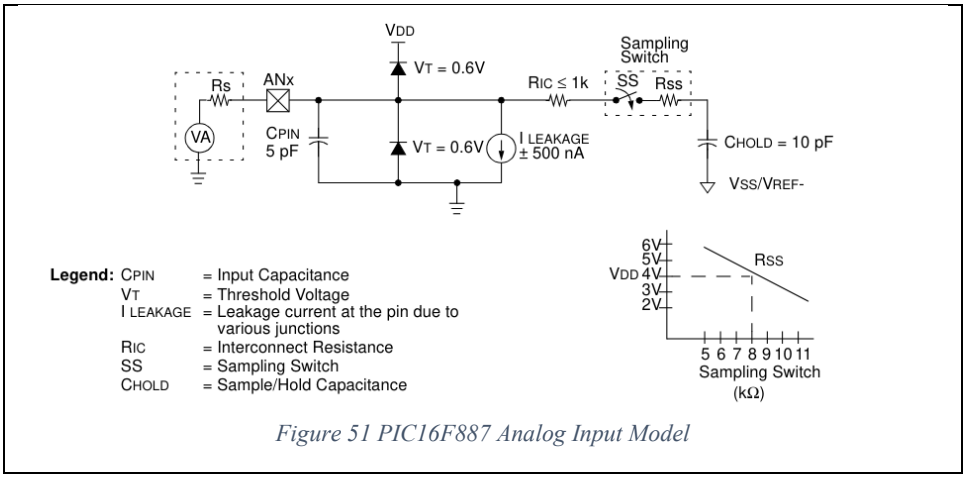


Figure 51 PIC16F887 Analog Input Model

3.4 DS1307 RTC

3.4.1 Electrical Specifications of DS1307

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Logic 1 Input	V_{IH}		2.2		V_{CC} + 0.3	V
Logic 0 Input	V_{IL}		-0.3		+0.8	V
V_{BAT} Battery Voltage	V_{BAT}		2.0	3	3.5	V

Figure 52 DS1307 recommended operating conditions

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I_{LI}		-1		1	µA
I/O Leakage (SDA, SQW/OUT)	I_{LO}		-1		1	µA
Logic 0 Output (I_{OL} = 5mA)	V_{OL}				0.4	V
Active Supply Current (f_{SCL} = 100kHz)	I_{CCA}				1.5	mA
Standby Current	I_{CCS}	(Note 3)			200	µA
V_{BAT} Leakage Current	I_{BATLKG}			5	50	nA
Power-Fail Voltage (V_{BAT} = 3.0V)	V_{PF}		1.216 x V_{BAT}	1.25 x V_{BAT}	1.284 x V_{BAT}	V

Figure 53 DS1307 DC electrical characteristics (on supply)

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{BAT} Current (OSC ON); SQW/OUT OFF	I_{BAT1}			300	500	nA
V_{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I_{BAT2}			480	800	nA
V_{BAT} Data-Retention Current (Oscillator Off)	I_{BATDR}			10	100	nA

Figure 54 DS1307 DC electrical characteristics (on battery)

AC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t_{BUF}		4.7			µs
Hold Time (Repeated) START Condition	$t_{HD:STA}$	(Note 4)	4.0			µs
LOW Period of SCL Clock	t_{LOW}		4.7			µs
HIGH Period of SCL Clock	t_{HIGH}		4.0			µs
Setup Time for a Repeated START Condition	$t_{SU:STA}$		4.7			µs
Data Hold Time	$t_{HD:DAT}$		0			µs
Data Setup Time	$t_{SU:DAT}$	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t_R				1000	ns
Fall Time of Both SDA and SCL Signals	t_F				300	ns
Setup Time for STOP Condition	$t_{SU:STO}$		4.7			µs

Figure 55 DS1307 AC characteristics

CAPACITANCE

(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	$C_{I/O}$				10	pF
Capacitance Load for Each Bus Line	C_B	(Note 7)			400	pF

Figure 56 DS1307 Pin capacitance

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate.

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_O		32.768		kHz
Series Resistance	ESR			45	kΩ
Load Capacitance	C_L		12.5		pF

Figure 57 DS1307 Crystal Specifications

3.4.2 I2C pull-up resistor calculation

The value of the pullup resistor is an important design consideration for I2C systems as an incorrect value can lead to signal loss. A low value resistor (strong pull-up) can prevent the microcontroller or the IC from pulling down the line at the appropriate time. On the other hand a high value resistor (weak pull-up) can prevent the line from going high within the stipulated amount of time.

According to the Texas Instruments I2C specifications, the pull-up resistor minimum value Rp(min) is given as

$$R_p(min) = (Vcc - V_{OL}(max))/I_{OL}$$

Calculating the Rp(min) considering the maximum current sink by PIC16F887 is about 25mA while DS1307 I2C pin is at 5mA with VOL = 0.4V, Rp(min) = (5 – 0.4)/5mA = 920Ω to 1kΩ approximate.

Pull up resistor maximum value Rp(max) is given as

$$Rp(max) = t_{rise}/(Ln((Vcc - V_{IL})/(Vcc - V_{IH})) \times C_{bus})$$

Considering the Bus Capacitance is 400pF, $V_{IH\ PIC16F887}$ = 2V and $V_{IH\ DS1307}$ = 2.2V and $V_{IL\ PIC16F887}$ = 0.8V and $V_{IL\ DS1307}$ = 0.8V, the Rp(max) is given as Rp(max) = 1000ns/(Ln((5 – 0.8)/(5 – 2.2)) x 400pF) = 6.165kΩ. Minimum power dissipation considering 50% duty cycle is therefore given as

$P_{RPULL\ dissipate} = 0.5 \times V^2/R = 0.5 \times 5^2/4.7 = 2.65mW$ with total dissipation on both lines together is 5.3mW. A nominal 4.7kΩ 1/8 Watt resistor can therefore be considered.

HW111 DS1307 module uses 3.3kΩ pull-up resistors. So, power dissipation will be $0.5 \times 5^2/3.3$ = 3.78mW and total dissipation on both lines will be 7.56mW.

3.5 Using 3.3V regulator for powering ESP8266-01

Unlike other components in the project which can work on 5V, ESP-01 is 3.3V tolerant. Therefore, for powering this Wi-Fi MCU a regulator with at least 500mA current output capability is needed. AMS1117 is used for regulating 5V to 3.3V and powering the ESP-01.

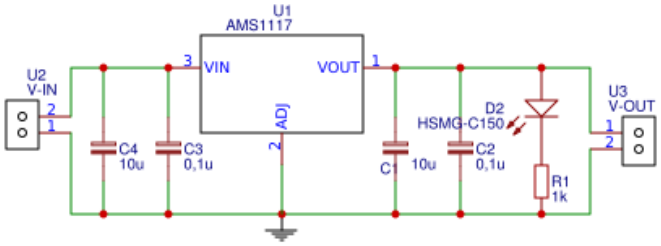


Figure 58 AMS1117-3.3V Schematic

AMS1117 has a Quiescent Current of 11mA maximum. The dissipation of power as heat is given as (5-3.3) * 300mA = 0.51W while power consumption of ESP-01 is given as 3.3 * 300mA = 0.99W.

AMS1117 General Specification

Output Type	Fixed
Polarity	Positive
Input Voltage Range	1.8V- 15V DC
Output Voltage(V)	3.3
Max. Output Current (A)	1

3.6 Power Specification of Entire Project

A USB power source (5V) is used to power the entire project. Total current consumption of the entire project can be calculated by considering the current consumption of individual components which is given in the table below.

Values below are taken from the Datasheet of the respective components.

Component Name	**Current Consumption**
PIC16F887	HFINTOSC 4Mhz: 1.6mA SLEEP LED Current Source: 0.5mA SW0 and SW1 Weak Pull up: 0.8mA ADC (Not always on): 1mA during initial acquisition. 0.05mA during A/D Conversion cycle I2C lines can consume current up to 1.5mA. SPI lines have consumption of about 1µA for DIN, CLK, LOAD It can be therefore considered that maximum current consumption in worst case will be 10mA (Assumed) for PIC16F887.
DS1307	Max current consumption is not more than 2mA. SQOut pin is pulled up by 3.3kΩ with max current consumption of 1.5mA.
MAX7219	Maximum supply current consumption is 50mA (according to the 0.248W dissipation calculated in Page 33) (4 digits and all segments turned on)
ESP-01	Average Operating Current is 80mA but may peak up to 300mA during Wi-Fi operations.
AMS1117	Quiescent current: 11mA maximum

Maximum current can therefore be not more than 500mA. A typical USB source with 500mA current output can be used. However, it is recommended to use a source with about 800mA current handling capacity as ESP8266 tends to spike up in current consumption while performing Wi-Fi operations.

4 References

- A Beginner's Guide to the ESP8266
  https://tttapa.github.io/ESP8266/Chap01 - ESP8266.html

- MAX7219/MAX7221 Serially Interfaced, 8-Digit LED Display Drivers
  https://www.analog.com/media/en/technical-documentation/data-sheets/max7219-max7221.pdf

- PIC16F882/883/884/886/887 Data Sheet
  ww1.microchip.com/downloads/en/devicedoc/41291d.pdf

- DS1307
  https://www.analog.com/media/en/technical-documentation/data-sheets/DS1307.pdf

- DS1307 Breakout Board Schematic
  https://forum.arduino.cc/t/tiny-rtc-charging-circuit-schematic/431613

- AMS1117
  http://www.advanced-monolithic.com/pdf/ds1117.pdf

- Understanding I2C Bus
  https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1677309272009&ref_url=https%253A%252F%252Fwww.google.com%252F#:~:text=The_I2C_bus_is_a_standard_bidirectional_interface_that_uses,been addressed by the master.

- I2C Bus Pullup Resistor Calculation
  https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/6/I2C-Bus-Pullup-Resistor-Calculation.pdf

- Bi-Directional Converter Guide
  https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide