EMBEDDED REPORT

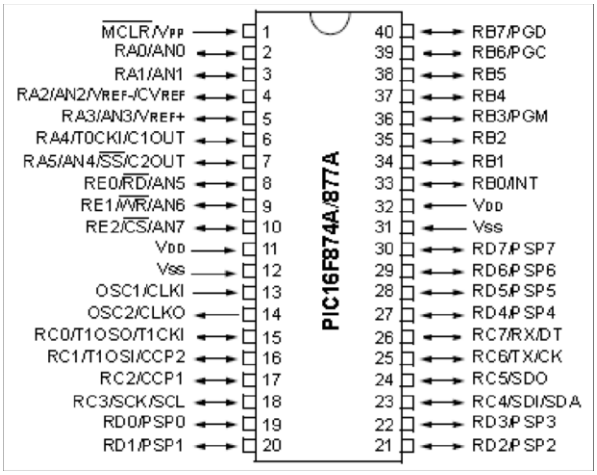
Traffic Light Controller

MINA GAMIL
B.N: 52
2nd year communications and computer science

PIC16F877A

Description:PIC16F877A is one of microchip 8-bit microcontrollers and the most popular one that is used in many applications such as robotics, embedded systems, and industrial automation due to its versatility and robust feature set. The term of PIC16F indicates that it belongs to the PIC16 series, which is classified as mid-range in Microchip's lineup of microcontrollers, the PIC16 series includes microcontrollers with 8-bit data paths and a 14-bit wide instruction word, which makes it suitable for a wide range of applications with moderate complexity.

Task1:



a) Power Supply Pins
   1. **VDD (Pins 11 and 32)**: Positive supply for logic and I/O pins.
   2. **VSS (Pins 12 and 31)**: Ground reference for logic and I/O pins.

   Oscillator Pins

   3. **OSC1/CLKIN (Pin 13)**: Oscillator crystal or external clock input.
   4. **OSC2/CLKOUT (Pin 14)**: Oscillator crystal or clock output.

   Reset Pin

   5. **MCLR/Vpp (Pin 1)**: Master Clear (input) or programming voltage (output).

I/O Port Pins

Port A

6. **RA0/AN0 (Pin 2)**: Digital I/O, Analog input 0.
7. **RA1/AN1 (Pin 3)**: Digital I/O, Analog input 1.
8. **RA2/AN2/VREF-/CVREF (Pin 4)**: Digital I/O, Analog input 2, A/D reference voltage (Low) input, Comparator VREF output.
9. **RA3/AN3/VREF+ (Pin 5)**: Digital I/O, Analog input 3, A/D reference voltage (High) input.
10. **RA4/T0CKI/C1OUT (Pin 6)**: Digital I/O – Open-drain when configured as output, Timer0 external clock input, Comparator 1 output.
11. **RA5/AN4/SS/C2OUT (Pin 7)**: Digital I/O, Analog input 4, SPI slave select input, Comparator 2 output.

Port B

12. **RB0/INT (Pin 33)**: Digital I/O, External interrupt.
13. **RB1 (Pin 34)**: Digital I/O.
14. **RB2 (Pin 35)**: Digital I/O.
15. **RB3/PGM (Pin 36)**: Digital I/O, Low-voltage ICSP programming enable pin.
16. **RB4 (Pin 37)**: Digital I/O.
17. **RB5 (Pin 38)**: Digital I/O.
18. **RB6/PGC (Pin 39)**: Digital I/O, In-circuit debugger and ICSP programming clock.
19. **RB7/PGD (Pin 40)**: Digital I/O, In-circuit debugger and ICSP programming data.

Port C

20. **RC0/T1OSO/T1CKI (Pin 15)**: Digital I/O, Timer1 oscillator output, Timer1 external clock input.
21. **RC1/T1OSI/CCP2 (Pin 16)**: Digital I/O, Timer1 oscillator input, Capture2 input, Compare2 output, PWM2 output.
22. **RC2/CCP1 (Pin 17)**: Digital I/O, Capture1 input, Compare1 output, PWM1 output.

23. **RC3/SCK/SCL (Pin 18)**: Digital I/O, Synchronous serial clock input/output for SPI mode, Synchronous serial clock input/output for I2C mode.
24. **RC4/SDI/SDA (Pin 23)**: Digital I/O, SPI data in, I2C data I/O.
25. **RC5/SDO (Pin 24)**: Digital I/O, SPI data out.
26. **RC6/TX/CK (Pin 25)**: Digital I/O, USART asynchronous transmit, USART1 synchronous clock.
27. **RC7/RX/DT (Pin 26)**: Digital I/O, USART asynchronous receive, USART synchronous data.

Port D

28. **RD0/PSP0 (Pin 19)**: Digital I/O, Parallel Slave Port data.
29. **RD1/PSP1 (Pin 20)**: Digital I/O, Parallel Slave Port data.
30. **RD2/PSP2 (Pin 21)**: Digital I/O, Parallel Slave Port data.
31. **RD3/PSP3 (Pin 22)**: Digital I/O, Parallel Slave Port data.
32. **RD4/PSP4 (Pin 27)**: Digital I/O, Parallel Slave Port data.
33. **RD5/PSP5 (Pin 28)**: Digital I/O, Parallel Slave Port data.
34. **RD6/PSP6 (Pin 29)**: Digital I/O, Parallel Slave Port data.
35. **RD7/PSP7 (Pin 30)**: Digital I/O, Parallel Slave Port data.

Port E

36. **RE0/RD/AN5 (Pin 8)**: Digital I/O, Read control for Parallel Slave Port, Analog input 5.
37. **RE1/WR/AN6 (Pin 9)**: Digital I/O, Write control for Parallel Slave Port, Analog input 6.
38. **RE2/CS/AN7 (Pin 10)**: Digital I/O, Chip select control for Parallel Slave Port, Analog input 7.

b)

1. **Arithmetic Logic Unit (ALU)**: The ALU performs arithmetic and logical operations. It is capable of handling operations like addition, subtraction, bitwise AND, OR, XOR, increment, decrement, and various shift operations.

2. **Status and Control**: This block includes the STATUS register and other control registers that monitor and control the operation of the microcontroller. The Status register contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory. The Status register can be the destination for any instruction, as with any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable, therefore, the result of an instruction with the Status register as destination may be different than intended.
3. **Program Counter (PC)**: The program counter holds the address of the next instruction to be executed from the Flash program memory. Ensures the sequential execution of instructions. It increments after fetching each instruction, but can also be modified by jump, call, and return instructions to change the execution flow.
4. **Flash Program Memory**: The flash program memory is readable and writable during normal operation (over the full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are six SFRs used to read and write this memory:
    - EECON1
    - EECON2
    - EEDATA
    - EEDATH
    - EEADR
    - EEADRH
5. **Instruction Register**: It's also called buffer register, temporarily holds the current instruction fetched from the program memory before it is decoded and executed.
6. **Instruction Decoder**: Interprets the instruction held in the Instruction Register and generates the necessary control signals to execute it, converts the binary instruction code into a series of control signals that

drive the various parts of the microcontroller to perform the specified operation.

c) The RA4/T0CKI pin is a Schmitt Trigger input and an open-drain output. In an open-drain configuration, the output transistor can pull the pin to ground (logic 0) but cannot drive it to a high level (logic 1). To achieve a high level, an external pull-up resistor is needed to pull the pin up to the supply voltage (VDD). All other PORTA pins have TTL (Transistor-Transistor Logic) input levels and full CMOS output drivers.

d)

1. Memory Size
    - Program Memory (Flash):
        - **ATMega328P**: 32 KB
        - **PIC16F877A**: 14 KB
    - RAM:
        - **ATMega328P**: 2 KB
        - **PIC16F877A**: 368 bytes
    - EEPROM:
        - **ATMega328P**: 1 KB
        - **PIC16F877A**: 256 bytes
2. Power Consumption
    - Operating Voltage:
        - **ATMega328P**: 1.8V to 5.5V
        - **PIC16F877A**: 2.0V to 5.5V
    - Power Consumption:
        - **ATMega328P**: Typically lower power consumption in both active and sleep modes compared to PIC16F877A.
        - **PIC16F877A**: Higher power consumption in comparison.
3. Pin Count

- **ATMega328P**: 28 pins (PDIP)
- **PIC16F877A**: 40 pins (PDIP)

4. Clock Speed
    - **ATMega328P**: Up to 20 MHz
    - **PIC16F877A**: Up to 20 MHz
5. Peripheral Features
    - **ATMega328P**:
        - 1x 16-bit Timer/Counter
        - 2x 8-bit Timer/Counters
        - 6-channel 10-bit ADC
        - USART, SPI, I2C (TWI)
        - PWM channels
    - **PIC16F877A**:
        - 3x Timers (one 8-bit, two 16-bit)
        - 8-channel 10-bit ADC
        - USART, SPI, I2C
        - PWM channels
        - CCP (Capture/Compare/PWM) modules
6. Development Ecosystem
    - **ATMega328P**:
        - Widely supported by the Arduino ecosystem, making it very beginner-friendly.
        - Extensive community support and libraries.
    - **PIC16F877A**:
        - Supported by Microchip's MPLAB ecosystem.
        - Good community support but generally considered less beginner-friendly compared to Arduino.

Examples of Embedded Systems Where ATMega328P is a Better Choice:

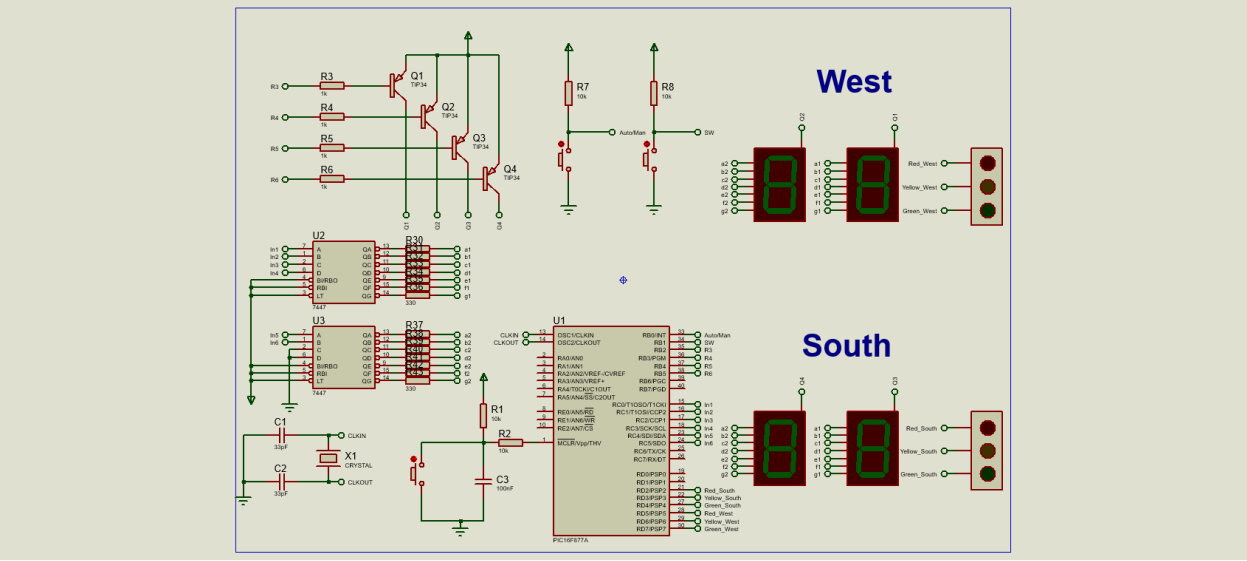
1. Arduino-Based Projects:
    - Reason: The ATMega328P is the core of the Arduino Uno board, which has extensive support, libraries, and documentation. It is ideal

for rapid prototyping and projects where ease of use and community support are critical.

- Example: A DIY home automation system using various sensors and actuators, where quick development and a plethora of ready-made libraries are advantageous.

2. Battery-Powered IoT Devices:
    - Reason: Due to its lower power consumption and sleep modes, the ATMega328P is better suited for battery-powered applications. Its efficiency in sleep mode extends battery life significantly.
    - Example: A remote environmental monitoring system with temperature, humidity, and pressure sensors, designed to run for extended periods on battery power.

Task2:



Review about the circuit:

- ❖ Four 7-segment common anodes, two for West Street and the others for South Street. One of each of them is used for indicating ones, and the other of each of them is used for indicating tens. The two that are used for ones their pins are connected to the same decoder (7447), and the others with another decoder of the same type. Each one of them its pin of common

anode is connected with a transistor which when its base is supplied with a low voltage (logic 0) then the 7-segment can be turned on, and when supplied with a high voltage (logic 1) can be turned off.

- The decoders are used for converting a binary code to a decimal code (BCD) and their pins are connected to C pins.
- Each of the streets has a traffic light, and the pins of each of them are connected to D pins.
- The sides that supply the base of each of transistors are connected to B pins.
- 2 switches are connected to B pins, one of them is connected to RB0 which can trigger interrupt in the falling level when the push button is pressed to switch from automatic mode to manual mode. The other switch is used for switching between the two streets in the manual mode.

Review about the code:

- There four variables are assigned as flags, the (flag) flag is assigned to break the while loop in interrupt function, the (West_flag) flag is used for ensuring the code when is switched from manual to automatic mode that it came from the South area (equal to 0) or West area (equal to 1), the (counter) flag is used for ensuring that the code has been through the area of South in interrupt function at least one time and is used with West_flag because in the beginning West_flag is 0 so it can break the west function so to ensure that can't be happen there is the counter flag, the (South_flag) flag is used for ensuring the code when is switched from automatic to manual mode that it came from the South area (equal to 1) or West area (equal to 0).
- An array called segment contains hex numbers from 0 to 23 to use them to show the numbers on 7-segments in the for loop of west function(20s green and 3s yellow of West, 23s red of South) and south function (12s green and 3s yellow of South, 15s red of West).
- south_yellow function to turn on the yellow of south and the red of west, south_green function to turn on the green of south and the red of west, west_yellow function to turn on the red of south and the yellow of west, west_green function to turn on the red of south and the green of west.

- Interrupt function is used in the manual mode with infinite while loop and continues from the triggered point (where the code has been triggered interrupt) by South_flag (if 1 then it continues with south_green, if 0 then it continues with west_green) and switches between the streets by the other switch (SW) and when the Auto/Man switch is pressed it returns back and continues from the stop point in manual mode by West_flag (if 1 then it continues with west function in the main, if 0 then it continues with south function and the counter flag to check that the code has been through the area of South in interrupt function at least one time) and to ensure that there's no error occurred the INTE_bit is set to zero with INTF_bit and when the while loop in interrupt has been broken, afterwards that, it will set again to 1.
- west function to turn on the green light 20s and the yellow light 3s each of West Street while the red light of South Street is turned on 23s.
- south function to turn on the green light 12s and the yellow light 3s each of South Street while the red light of West Street is turned on 15s.

Click here to see the project: Traffic Light Controller (https://www.youtube.com/watch?v=Tye5bUF7SSg&t=21s)